

**CONTROL PARA SISTEMAS CONTINUOS Y DISCRETOS BASADO EN  
LA INGENIERÍA DIRIGIDA POR MODELOS**

**CONTROL FOR CONTINUOUS AND DISCRETE SYSTEMS BASED ON MODEL  
DRIVEN ENGINEERING**

**ANDRÉS FELIPE ERAZO RAMOS**

Asesora

**PAOLA ANDREA VALLEJO CORREA**

**REQUISITO PARA OPTAR AL TÍTULO  
DE MAGÍSTER EN INGENIERÍA**

**UNIVERSIDAD EAFIT  
ESCUELA DE INGENIERÍA  
MEDELLÍN - ANTIOQUIA - COLOMBIA  
2020**

## Resumen

El control ha sido un factor fundamental para el desarrollo de la tecnología, siempre que se requiere ajustar una variable física a un valor específico se está realizando una acción de control.

El control se puede encontrar en la vida cotidiana en actividades como ajustar la temperatura de la ducha o ajustar el nivel de luminosidad en un invernadero. En términos generales, un sistema de control es un componente de un sistema más complejo en el que se requiere producir una salida determinada o mantener un estado específico del sistema. Los sistemas que requieren de un proceso de control retroalimentado son cada vez más complejos, lo cual conlleva grandes desafíos. La presente investigación tiene como propósito proponer un enfoque de simple utilización, basado en la ingeniería dirigida por modelos para el control de sistemas continuos y sistemas con eventos discretos, que incluye la implementación de la propuesta en la plataforma VariaMos y su aplicación en dos pruebas de concepto. La implementación permite, de manera experimental, obtener el modelo analítico del sistema y las ganancias del controlador PID. Los resultados obtenidos evidencian la manera en la que el enfoque puede ser usado en sistemas con eventos continuos y discretos.

Por lo tanto, se concluye que el enfoque propuesto permite producir sistemas con capacidades de retroalimentación. Además, la implementación del enfoque facilita su uso a usuarios que no son necesariamente expertos en control.

## **Abstract**

The control has been a fundamental factor for the development of the technology, whenever a physical variable needs to be adjusted to a specific value, a control action is being performed. Control can be found in everyday life in activities like adjusting the temperature of the shower or adjusting the brightness in a greenhouse. In general terms, a control system is a component of a more complex system in which it is required to produce a certain output or maintain a specific system state. Systems that require a feedback control process are becoming increasingly complex, which carries with it great challenges.

This research aims to propose a simple model-driven engineering approach, for the control of continuous systems and discrete-event systems, which includes the implementation of the proposal on the VariaMos platform and its application in two proof of concept. The implementation allows in an experimental way, to obtain the analytical model of the system and the gains of the PID controller. The results obtained show how approach can be used in systems with continuous and discrete events.

Therefore, it is concluded that the proposed approach allows producing systems with feedback capabilities. Furthermore, the implementation of the approach allows it to be used by users who are not necessarily experts in control.

## Tabla de Contenidos

Lista de figuras .....	vi
Capítulo 1 Introducción.....	1
1.1 Motivación .....	1
1.2 Planteamiento del problema.....	2
1.3 Preguntas de investigación.....	3
1.4 Hipótesis de investigación .....	3
1.5 Metodología de investigación .....	3
1.6 Resultados.....	5
1.7 Contribuciones y conclusiones.....	5
1.8 Organización del documento .....	7
Capítulo 2 Conceptos clave .....	8
Capítulo 3 Estado del arte .....	22
Capítulo 4 Ejemplos.....	27
4.1 Sistema de iluminación.....	27
4.1.1 Descripción del sistema. ....	27
4.1.2 Arquitectura e implementación del sistema. ....	29
4.2 Sistema de venta virtual de prendas de vestir .....	31
4.2.1 Descripción del sistema. ....	31
4.2.2 Arquitectura e implementación del sistema. ....	32
Capítulo 5 Resumen de la propuesta.....	36
Capítulo 6 Propuesta .....	39
6.1. Etapa preliminar .....	39
6.2 Etapa de control.....	39
6.3 Implementación.....	41
6.3.1 Metamodelo.....	43
6.3.2 Modelo. ....	46
6.3.3 Visión general.....	47
6.4 Aplicación en sistema continuo .....	49
6.4.1 Identificación del sistema.....	50
6.4.2 Estimación del sistema.....	55
6.4.3 Respuesta del sistema con controlador. ....	56
6.4.4 Simulación del sistema de iluminación.....	57
6.5 Aplicación en sistema discreto.....	58
6.5.1 Identificación del sistema.....	59
6.5.2 Estimación del sistema.....	60
6.5.3 Identificación del sistema.....	61
6.5.4 Simulación del sistema. ....	62
Capítulo 7 Evaluación de la propuesta.....	64
Capítulo 8 Resultados .....	69
8.1 Módulo de control .....	69

8.2 Evaluación de un sistema de ventas con controlador .....	69
8.3 Conexión Modulo de hardware VariaMos.....	70
Capítulo 9 Discusión y validez de los resultados .....	71
Capítulo 10 Conclusiones y trabajo a futuro .....	72
Lista de referencias .....	74
ANEXOS .....	78
ANEXO 1: Código del sistema de iluminación sin control.....	79
ANEXO 2: Código del sistema de iluminación con control .....	82
ANEXO 3: Código fuente tienda de ropa.....	86
3.1 Estructura de carpetas .....	86
3.2 Modelos .....	87
3.3 Controladores .....	87
3.4 Vista .....	87
ANEXO 4: VariaMos Web.....	88
4.1 Arquitectura .....	88
4.1.1 Vista del Cliente Modulo de control.....	89
ANEXO 5: Prueba de usabilidad .....	92
5.1 Métricas .....	94
5.2 Resultados .....	95
5.2.1 Efectividad .....	95
5.2.2 Operabilidad .....	96
5.2.3 Satisfacción .....	97

## Lista de figuras

Figura 1. Metodología de la ciencia del diseño aplicada a la investigación sobre la determinación experimental del mecanismo de control PID para sistemas de eventos continuos y discretos.	4
Figura 2. Sistema de control microondas.	9
Figura 3. Diagrama de bloques de un sistema retroalimentado.	10
Figura 4. Diagrama de bloques de un sistema de control de temperatura.	10
Figura 5. Dinámica del sistema.	11
Figura 6. Gráfica representativa del comportamiento discreto de un sistema. Tomado de (Dulces, 2013).	14
Figura 7. Estabilidad e inestabilidad de los sistemas.	14
Figura 8. Respuesta de un sistema de primer orden.	15
Figura 9. Respuestas posibles de los sistemas de segundo orden.	16
Figura 10. Respuesta del sistema con controlador. Tomado de (Hernández, 2015).	16
Figura 11. Curva reacción. Tomado de (Kandulna Ruben, 2015).	19
Figura 12. Ziegler Nichols lazo abierto. Adaptado de (Kushwah & Patra, 2014).	19
Figura 13. Curva reacción Cohen Coon. Tomado de (Badal et al., 2019).	20
Figura 14. Cohen Coon regla. Tomado de (Normey-Rico, 2007).	20
Figura 15. Amigo regla. Tomado de (Hagglund, 1995).	20
Figura 16. Sistema de iluminación.	28
Figura 17. Diseño electrónico sistema de iluminación.	29
Figura 18. Circuito electrónico sistema de iluminación.	29
Figura 19. Adquisición de datos sistema iluminación.	30
Figura 20. Menú inicial tienda ropa.	31
Figura 21. Menú lista de compras.	32
Figura 22. Arquitectura sistema de ventas.	33
Figura 23. Número de ventas vs Porcentaje de descuento.	35
Figura 24. Enfoque control retroalimentado.	37
Figura 25. Árbol de trabajo VariaMos.	42
Figura 26. Menú control.	43
Figura 27. Metamodelo sistema de control.	45
Figura 28. Bloques sistema de control.	46
Figura 29. Relaciones sistema de control.	46
Figura 30. Punto de suma.	46
Figura 31. Punto de ramificación.	47
Figura 32. Visión general en un sistema continuo.	48
Figura 33. Visión general en un sistema discreto.	49
Figura 34. Comportamiento sistema de iluminación.	50
Figura 35. Diagrama del bloque sistema de iluminación.	51
Figura 36. Setpoint sistema de iluminación.	51
Figura 37. Controlador sistema de iluminación.	51

Figura 38. Punto de suma sistema de iluminación.....	52
Figura 39. Sensor sistema de iluminación. ....	52
Figura 40. Salida del sistema de iluminación. ....	53
Figura 41. Acción de control Sistema de iluminación. ....	53
Figura 42. Planta sistema de iluminación.....	54
Figura 43. Identificación sistema de iluminación. ....	54
Figura 44. Estimación sistema de iluminación. ....	56
Figura 45. Salida sistema de iluminación con control. ....	57
Figura 46. Simulación del sistema de iluminación con control. ....	58
Figura 47. Sistema de ventas. ....	59
Figura 48. Identificación del sistema de ventas. ....	60
Figura 49. Estimación del sistema de ventas. ....	61
Figura 50. Respuesta del controlador en el sistema de ventas.....	62
Figura 51 Simulación sistema de ventas con control. ....	63
Figura 52. Evaluación sistema iluminación sin control vs sistemas iluminación con control. ....	65
Figura 53. Polos y Ceros en el sistema de Iluminación. ....	67
Figura 54. Evaluación sistema de ventas.....	68

## **Capítulo 1**

### **Introducción**

#### **1.1 Motivación**

Los controles automáticos tienen una influencia cada vez más importante en la vida cotidiana y en los procesos industriales y de manufactura modernos. En la actualidad se evidencia cada vez más la necesidad de usar sistemas de control para mejorar y optimizar procesos, en los que la intervención humana no es suficiente o no es recomendada. La implementación de sistemas de control retroalimentado en el mundo físico comprende un proceso exitoso, ya sea desde sistemas industriales, transporte, desarrollo de circuitos, entre otros. Su capacidad de autocorregirse con requisitos mínimos ante circunstancias cambiantes es un gran aporte, sin embargo, el aporte en el control de sistemas con eventos discretos aún está en una fase preliminar, especialmente en el control de sistemas de software.

Este enfoque reutiliza el conocimiento explícito del proceso de control utilizando las capacidades de un proceso retroalimentado con el fin de producir sistemas continuos y discretos controlados por un controlador PID (proporcional, integrativo y derivativo). El enfoque fue evaluado mediante el uso de pruebas de concepto correspondientes a un sistema de eventos continuos y a un sistema de eventos discretos.



## 1.2 Planteamiento del problema

En los últimos años se ha evidenciado un creciente interés en el estudio del control retroalimentado en sistemas computacionales, clasificados en sistemas continuos, por ejemplo, los procesos físicos, los cuales se describen matemáticamente por ecuaciones diferenciales; o sistemas discretos descritos por ecuaciones de diferencia, por ejemplo, en un restaurante una variable discreta es el tiempo promedio de espera, el registro del tiempo se hace por intervalos de tiempo, no de manera continua.

La metodología que parte del diseño del controlador para los sistemas discretos implica un alto grado de complejidad, en cuanto a la obtención del modelo matemático del sistema se refiere, este modelo constituye la base del proceso de control de un sistema retroalimentado, que permite evaluar la estabilidad, rendimiento y precisión de un sistema con control. Por ende, la participación de un experto en control generalmente es necesaria, además, hasta la fecha las contribuciones en el control de sistemas con eventos discretos son mínimas y generalmente limitadas a un enfoque teórico.

En este contexto, nos concentramos en analizar en qué medida un proceso de simple utilización, basado en la ingeniería dirigida por modelos permite aplicar principios de un sistema de retroalimentación con controlador PID en sistemas con eventos continuos y discretos.

### **1.3 Preguntas de investigación**

¿En qué medida las capacidades del enfoque pueden ser usadas para determinar el controlador PID de sistemas con eventos continuos y de sistemas con eventos discretos?

¿De qué manera influye el uso de principios básicos de un sistema de control retroalimentado en un sistema con eventos discretos?

¿Es suficiente el uso de reglas heurísticas para el ajuste de un controlador PID en un sistema con eventos discretos?

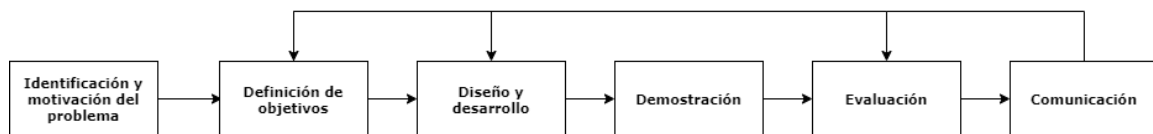
### **1.4 Hipótesis de investigación**

Es posible determinar el control PID en sistemas de eventos continuos y discretos mediante un proceso común y de simple utilización, basado en la ingeniería dirigida por modelos.

### **1.5 Metodología de investigación**

El documento actual es el resumen de un proyecto de investigación que tuvo como objetivo investigar el alcance de las capacidades de un sistema de control de retroalimentación en sistemas continuos y discretos. Para abordar las preguntas de investigación y probar la hipótesis de investigación, se utilizó una estrategia de investigación basada en la metodología de investigación llamada ciencia del diseño (o design science) propuesta por Peffers et al. (2008). Esta metodología se compone de seis etapas principales (ver Figura 1):

1. Identificación y motivación del problema.
2. Definición de objetivos.
3. Diseño y desarrollo.
4. Demostración.
5. Evaluación.
6. Comunicación.



*Figura 1. Metodología de la ciencia del diseño aplicada a la investigación sobre la determinación experimental del mecanismo de control PID para sistemas de eventos continuos y discretos.*

La etapa de identificación y motivación del problema se llevó a cabo de manera sistemática mediante una recopilación detallada de investigaciones preliminares, publicadas por distintos autores, entre las que se destacan las de Janert (2013) y Platzer (2012). En esta etapa se identificó la problemática de ¿En qué medida las capacidades de un sistema de control retroalimentado pueden incorporarse en sistemas continuos y sistemas con eventos discretos? En la etapa de definición de objetivos se definieron los siguientes objetivos:

- Proponer un enfoque para aplicar el control retroalimentado en sistemas continuos y sistemas a eventos discretos basado en la ingeniería dirigida por modelos.
- Proponer una implementación para el enfoque mencionado anteriormente.

En la etapa de diseño y desarrollo se definió un enfoque para especificar y aplicar el control retroalimentado en sistemas continuos y sistemas a eventos discretos, los detalles de esta etapa se presentan en el capítulo 5.

La etapa de demostración se presenta en el capítulo 6, donde se detalla el enfoque, junto con su implementación en una herramienta, y se aplica a los casos de estudio seleccionados.

La etapa de evaluación se detalla en el capítulo 7, donde se valida la utilidad y el alcance del enfoque propuesto, a través de un experimento controlado con la simulación del sistema.

Finalmente, para la etapa de comunicación se propone el presente documento cuyo objetivo es el de comunicar el desarrollo de la propuesta y los resultados obtenidos.

## **1.6 Resultados**

- Implementación en VariaMos del proceso de control para sistemas con eventos continuos y sistemas con eventos discretos.
- Desarrollo de una librería PID para el control en sistemas discretos.
- Evaluación de un sistema de información con principios de un sistema de control retroalimentado.

## **1.7 Contribuciones y conclusiones**

En el presente trabajo de investigación se abordan aspectos claves relacionados al proceso de control en sistemas discretos, que podrían aplicarse por ejemplo, en sistemas

de información, los cuales presentan un escenario idóneo, donde es necesario optimizar recursos frente a entornos cambiantes.

Se propone un enfoque de control retroalimentado desde la etapa de adquisición de datos del sistema hasta la generación de las ganancias del controlador PID para sistemas con eventos continuos y sistemas con eventos discretos como una tienda de ropa, con el propósito de efectuar la evaluación y utilidad del proceso. El enfoque propuesto se implementó como un módulo de la plataforma VariaMos<sup>1</sup>, la cual facilita el modelado de sistemas desde diferentes perspectivas. VariaMos fue diseñada como una herramienta de modelado que incorpora diferentes lenguajes de modelado para especificar y analizar familias de sistemas, por ejemplo, sistemas (auto) adaptables y líneas de productos (dinámicos) que soportan varios tipos de modelos y marcos en una aplicación orientada a los gráficos y disponible desde cualquier navegador web. El módulo de control, integrado con el módulo de diseño de hardware, permite diseñar diversos tipos de sistemas con control PID.

El enfoque permite evidenciar cómo procesos básicos de sistemas de información pueden ser representados como sistemas de primer y segundo orden, con el fin de aplicar capacidades de control retroalimentado en dichos procesos, utilizando metodologías heurísticas postuladas por Ziegler-Nichols, Cohen Coon y Amigo para el diseño del

---

<sup>1</sup> <http://variamos.dis.eafit.edu.co/>

controlador y las capacidades de VariaMos para simular el sistema, generar el código del controlador PID del sistema y validar el buen funcionamiento del sistema controlado.

## **1.8 Organización del documento**

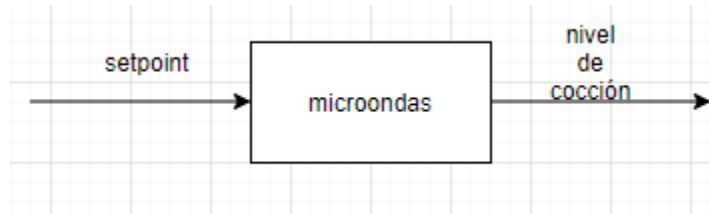
El documento está estructurado de la siguiente manera: El capítulo 2 se concentra en presentar los antecedentes y conceptos necesarios para comprender el desarrollo de la investigación. El capítulo 3 presenta los avances existentes en relación con el tema de investigación. El capítulo 4 presenta dos ejemplos que permitirán ilustrar el enfoque propuesto. El capítulo 5 presenta una visión global del enfoque, exponiendo los componentes necesarios para el desarrollo del proyecto. El capítulo 6 presenta en detalle el enfoque propuesto, así como los resultados obtenidos de aplicarlo en los ejemplos presentados en el capítulo 4, permitiendo validar su aplicación en sistemas con eventos continuos y sistemas con eventos discretos. El capítulo 7 presenta la evaluación de la propuesta. El capítulo 8 presenta los resultados principales obtenidos de la investigación. El capítulo 9 muestra el alcance de los resultados e informa sobre algunas amenazas a la validez de dichos resultados. El capítulo 10 concluye este informe y presenta perspectivas para futuros trabajos.

## Capítulo 2

### Conceptos clave

De forma informal, el **control** consiste en aplicar a un sistema fijo un conjunto de elementos (variables, valores, funciones, etc.) para que éste se comporte de una manera predefinida. Un **sistema de control** es definido como un conjunto de dispositivos interconectados que trabajan para lograr un mismo fin, capaces de regular el flujo de señales u otros recursos (Doyle et al., 2013). Un sistema de control puede ser tan simple como regular la temperatura del agua en una pecera o tan complejo como hacer que un robot cumpla un objetivo determinado.

Un sistema de control puede ser clasificado en sistema de lazo abierto o sistema de lazo cerrado; en los sistemas de **lazo abierto**, no hay retroalimentación constante en contraste con el comportamiento actual, es decir la salida no tiene efecto sobre la entrada del controlador (Berenguer, 2007). Un ejemplo común de sistema que opera en lazo abierto es el microondas, éste recibe como entrada el tiempo programado para la cocción de un alimento, y su salida es el nivel de cocción del alimento. En este caso el microondas no realiza retroalimentación sobre la salida. La Figura 2 representa el sistema de control del microondas.



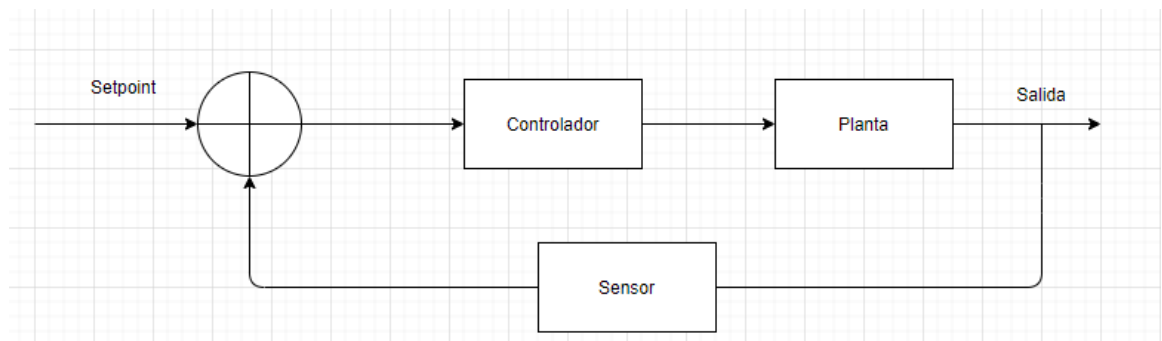
*Figura 2. Sistema de control microondas.*

Los sistemas de **lazo cerrado** o también denominados sistemas de control retroalimentado tienen como función operar constantemente al contrastar el estado actual de un sistema con su estado deseado. Si existe una discrepancia entre el comportamiento actual y el comportamiento proyectado, se aplica una acción correctiva hasta que el sistema alcanza su objetivo (Lobontiu, 2010). La Figura 3 representa un sistema de lazo cerrado, donde se encuentran los elementos más comunes como el valor de referencia o setpoint, el controlador del sistema, el sensor que captura el comportamiento del sistema y finalmente la salida del sistema. Para ilustrar este tipo de sistemas y la interacción entre los componentes generalmente se usan los diagramas de bloques. En un diagrama de bloques, la flecha que apunta hacia el bloque indica la señal de entrada y la flecha que sale del bloque indica la señal de salida. Las señales sólo pueden circular en el sentido en que indican las flechas.

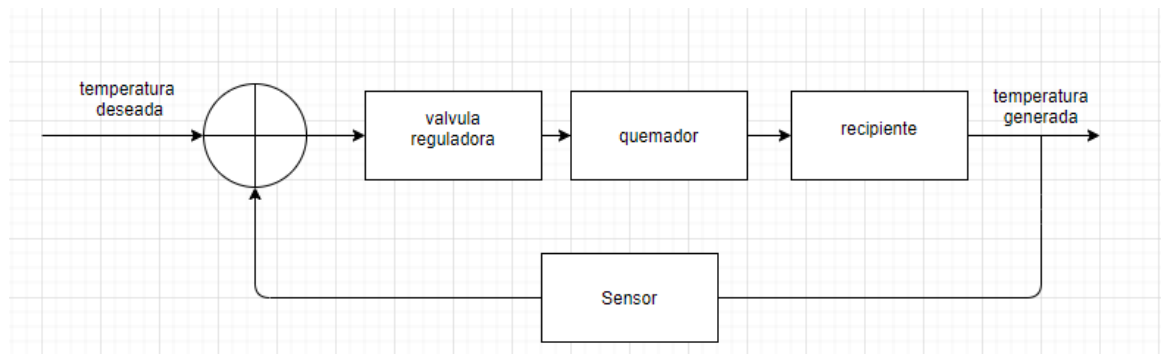
Un ejemplo de un sistema en lazo cerrado es el control de temperatura en el contenido de un recipiente, en el cual se tiene un valor de referencia o setpoint que es la temperatura deseada, se tiene el controlador que en este caso es una válvula reguladora, un actuador que es el quemador y una salida que es la temperatura generada. La Figura 4



ilustra mediante un diagrama de bloques el control de temperatura en el contenido de un recipiente.



*Figura 3. Diagrama de bloques de un sistema retroalimentado.*

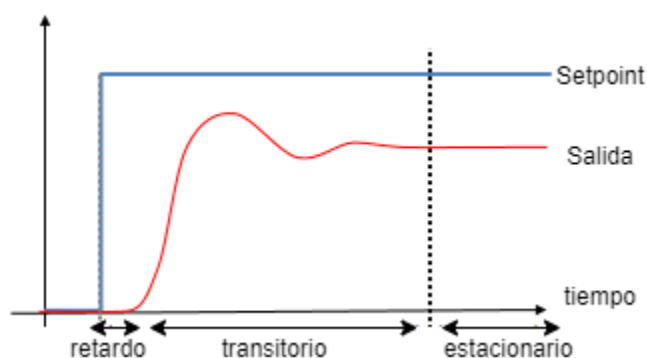


*Figura 4. Diagrama de bloques de un sistema de control de temperatura.*

A continuación, se presenta la terminología fundamental para la descripción de un sistema de control:

- **Planta:** Representa cualquier objeto físico que puede ser controlado (Simrock, 2011).
- **Setpoint:** Representa el punto objetivo del sistema, es decir qué valor de la variable física se desea alcanzar (Agbesi, 2006).

- **Controlador:** Es un componente diseñado para modular la dinámica de respuesta de un sistema, produciendo una acción de control que permita disminuir la diferencia entre el valor real y el valor deseado (Frank, 2018).
- **Salida:** Describe el valor actual del sistema o de la planta que está siendo controlada o monitoreada (Zeqiri & Luma, 2008). La salida del sistema a la respuesta de una entrada con función unitaria pasa por varios estados, como se presenta a continuación (ver Figura 5):



*Figura 5. Dinámica del sistema.*

- **Estado de retardo:** También llamado tiempo muerto, que representa el tiempo que tarda el sistema en responder a una entrada (Azamfar et al., 2016).
- **Estado transitorio:** La respuesta varía a lo largo del tiempo hasta llegar a un punto de estabilidad (Janert, 2013).
- **Estado estacionario:** Representa el comportamiento del sistema cuando alcanza un punto de estabilidad (Janert, 2013).

- **Error:** Es la diferencia entre el valor de referencia deseado (setpoint) y el valor actual del sistema (salida) (Christiansen et al., 2004).

$$E = \text{Setpoint} - \text{Salida}$$

- **Sensor:** Instrumento capaz de determinar una magnitud física, utilizado para medir el comportamiento de un sistema (Carrillo, 2008), por ejemplo, si deseamos controlar la temperatura, la salida puede medirse por un termopar.
- **Perturbaciones:** Representan las fuerzas externas que afectan el rendimiento del sistema (Carrillo, 2008).

Un sistema de lazo cerrado tiene un mecanismo de **retroalimentación** la cual se puede definir como información que regresa al lugar donde se originó; a partir de dicha información se desarrollan acciones correctivas. La retroalimentación permite tomar decisiones con respecto al cambio de alguna variable de entrada de un proceso, para cambiar el comportamiento de una variable de salida (Sharma, 2011). En la ingeniería de control se busca que las acciones correctivas se realicen de manera autónoma, es decir, por parte del mismo dispositivo, sin necesidad de que un humano realice manualmente la corrección. La realimentación se centra en características como: el desempeño del sistema, la estabilidad, la ganancia global, las perturbaciones y la sensibilidad. Cuando las variables comparadas son de igual signo se conoce como realimentación positiva. Si las variables comparadas son de signo contrario se conoce como realimentación negativa.

Además de los diagrama de bloques, las **funciones de transferencia** también permiten describir las relaciones entre las entradas y salidas en un sistema de control ya sea en lazo abierto o cerrado (Burghes & Graham, 2010).

Un sistema de control se puede clasificar según el tipo de comportamiento en las variables del sistema:

1. **Sistemas continuos:** “Las variables están definidas en todo instante de tiempo” (Llobregat et al., 2015), representadas por ecuaciones diferenciales o un operador lineal muy común como la transformada de Laplace. Por ejemplo, una aeronave que se desplaza en el aire, las variables como la velocidad y posición pueden cambiar instantáneamente con respecto al tiempo.
2. **Sistemas con eventos discretos:** “La modificación de los atributos se produce a través de eventos en instantes de tiempo determinados” (Llobregat et al., 2015). Por ejemplo, el promedio de clientes que llegan a un cine, sin importar el momento en el que lleguen, creando intervalos de tiempo para su registro. La Figura 6 representa la salida de un sistema con tiempo discreto donde sólo se toman valores en ciertos instantes de tiempo.

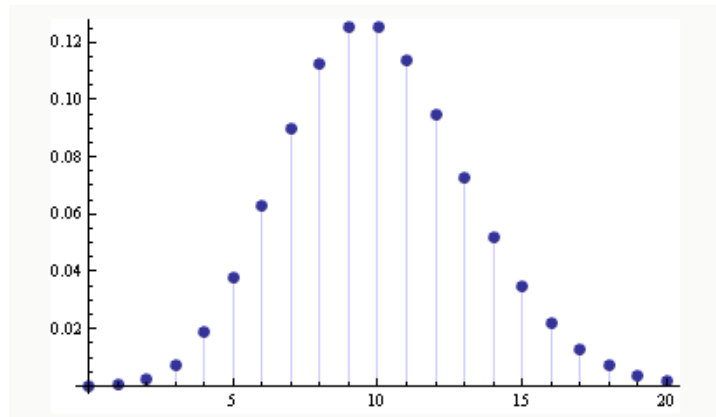


Figura 6. Gráfica representativa del comportamiento discreto de un sistema. Tomado de (Dulces, 2013)<sup>2</sup>

La respuesta de un sistema permite identificar si un sistema es **estable**, es decir permanece en un estado constante (Figura 7 izquierda). Un sistema es **inestable** cuando no se estabiliza en un valor constante, sino que este sigue aumentando (Figura 7 derecha).

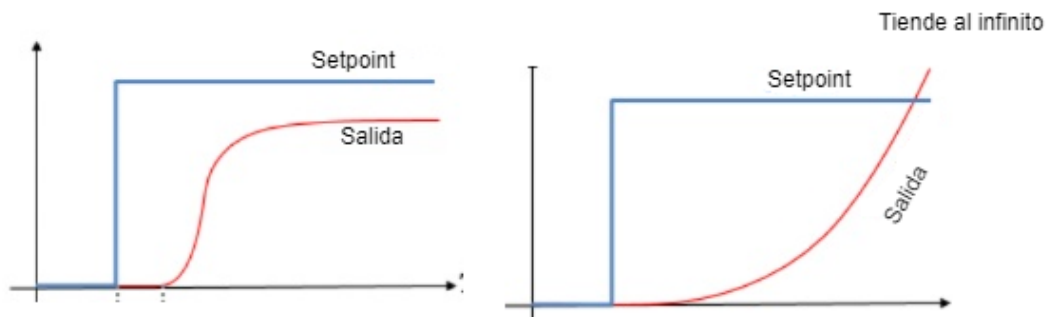
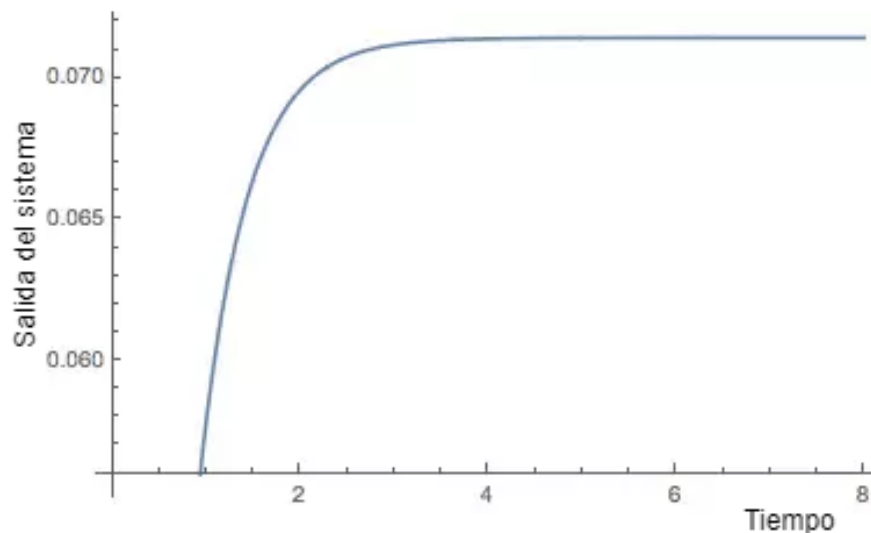


Figura 7. Estabilidad e inestabilidad de los sistemas.

---

<sup>2</sup> Recuperado de <https://samchappelle.wordpress.com/2013/08/25/basic-intro-to-poisson-random-variables/>

Adicional a lo anterior, los sistemas están clasificados según el orden, siendo los más representativos los sistemas de **primer orden**, aquellos sistemas cuya ecuación de entrada-salida es una ecuación diferencial de primer orden y los sistemas de **segundo orden**, representados por una ecuación diferencial de segundo orden. Los sistemas de segundo orden pueden responder de diferentes maneras: amortiguados, críticamente amortiguados o sobre-amortiguados. La Figura 8 representa la respuesta de un sistema de primer orden en la cual la dinámica no debe presentar amortiguaciones (sobre-impulsos) y la Figura 9, donde el eje horizontal indica el tiempo transcurrido y el eje vertical indica la salida del sistema, representa la dinámica de un sistema de segundo orden en los 3 estados posibles.



*Figura 8. Respuesta de un sistema de primer orden.*

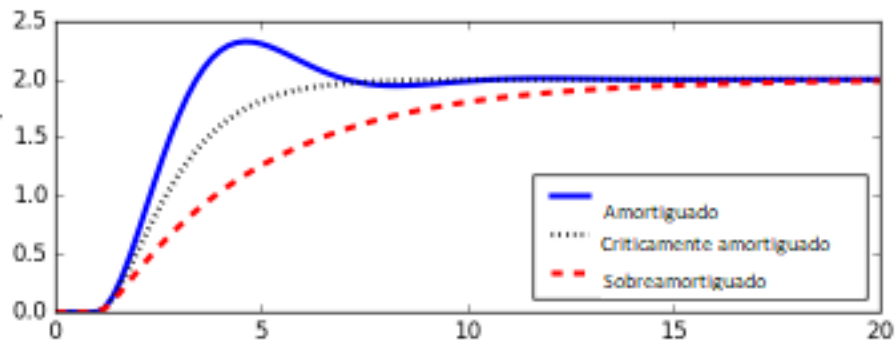


Figura 9. Respuestas posibles de los sistemas de segundo orden.

El objetivo de un sistema de control es lograr que la salida se acerque tanto como sea posible a un valor de referencia (setpoint), y que el error tienda a cero conforme el tiempo avance. La Figura 10 representa un sistema de temperatura con controlador, donde el sistema tiende a estabilizarse en un setpoint de  $40^{\circ}$ .

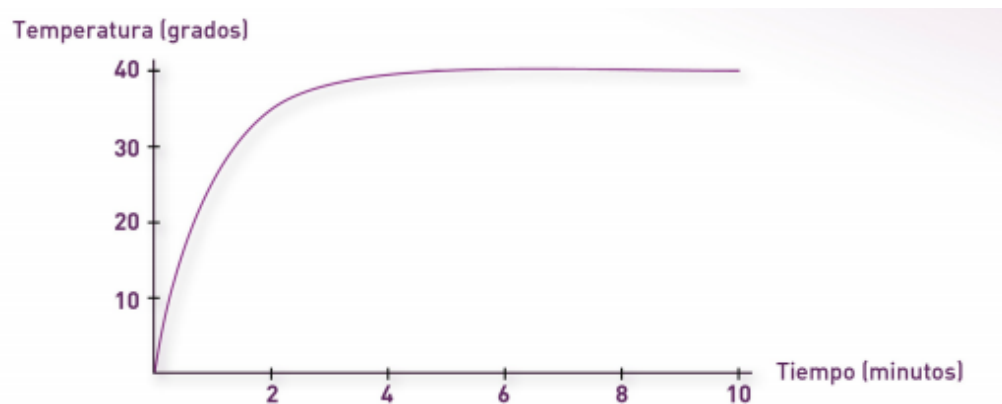


Figura 10. Respuesta del sistema con controlador. Tomado de (Hernández, 2015)

Existe una gran variedad de controladores, siendo el controlador **PID** el más común y utilizado. Este controlador está representado por tres parámetros: proporcional,

integral y derivativo, en el que cada usuario es libre de escoger los valores para llevar al sistema a un comportamiento deseado.

**Ganancia proporcional:** Representa el producto entre el error en un instante  $t$  ( $e(t)$ ) y una constante proporcional ( $K_p$ ). La fórmula que representa la salida ( $u(t)$ ) de una acción proporcional es:

$$u(t) = K_p * e(t) \quad (1)$$

Con  $K_p > 0$

**Ganancia integral:** Cuando se desea corregir un error en estado estacionario pequeño, no basta con aplicar una acción de control proporcional ya que la acción correctiva también será pequeña, para disminuir este error se integra los errores acumulados en el tiempo hasta reducir el error a cero y multiplicar por una constante ( $K_i$ ) (Visioli, 2006). La fórmula que representa la salida ( $u(t)$ ) de una acción integral es:

$$u(t) = K_i \int_0^T e(\tau) dt \quad (2)$$

Con  $K_i > 0$

**Ganancia derivativa:** Busca anticiparse a los acontecimientos, si la derivada del error es positiva, sabemos que el error de seguimiento está creciendo. Por lo tanto, se aplica una acción correctiva de inmediato, incluso si el valor del error aún es pequeño, para contrarrestar el crecimiento del error, antes de que empiece a incrementar (Janert, 2013). La fórmula que representa la salida ( $u(t)$ ) de una acción derivativa es:

$$u(t) = K_d \frac{de(t)}{dt} \quad (3)$$

Con  $K_d > 0$



Finalmente, la salida ( $u(t)$ ) de un controlador PID es la combinación de las tres ganancias:

$$u(t) = K_p \cdot e(t) + K_I \int_0^T e(t)dt + K_d \frac{de(t)}{dt} \quad (4)$$

Un cambio en las ganancias de un controlador afecta significativamente la salida de un sistema, por ende, es de suma importancia encontrar los valores adecuados del controlador para obtener una mejor respuesta, si se dispone de un modelo matemático del sistema se puede utilizar métodos como: posicionamiento de polos, para el ajuste de las ganancias del controlador PID, pero si el modelo analítico del sistema no se conoce o simplemente está incompleto, existen métodos experimentales que permite estimar los parámetros del controlador mediante la respuesta del sistema a una entrada determinada, los métodos más conocidos son los desarrollados por Ziegler-Nichols, Cohen Coon y Astrom-Hagglund (Amigo).

El **método de Ziegler-Nichols** (lazo abierto) también conocido como reacción de proceso, tiene la capacidad de probar la reacción de un sistema a un cambio en la variable de ajuste, es un método rápido y fácil de usar (Kushwah & Patra, 2014). Para aplicar este método el sistema debe estar en un estado estable o en un estado inicial, luego se modifica la variable de ajuste para cambiar la respuesta del sistema, si esta respuesta es de tipo sigmoidea es posible utilizar este método, ya que en este método se desea estimar el sistema a una función de primer orden, si es así se traza una tangente en el punto de inflexión de la curva para estimar los parámetros A y L (constante de tiempo) (ver Figura 11). Con estos parámetros podemos remplazar en las reglas propuestas por Ziegler para

calcular la ganancia proporcional ( $K$ ), tiempo integral ( $T_i$ ), tiempo derivativo ( $T_d$ ) (ver Figura 12). Este proceso se realiza en poco tiempo porque no hay necesidad de esperar a que la respuesta del sistema se estabilice, sin embargo, no es muy preciso.

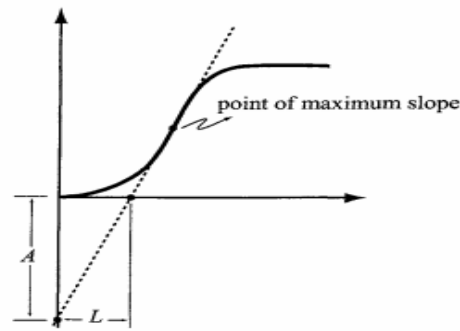


Figura 11. Curva reacción. Tomado de (Kandulna Ruben, 2015).

Tipo Control	$K$	$T_i$	$T_d$
P	$1/a$		
PD	$0.9/a$	$3L$	
PID	$1.2/a$	$2L$	$L/2$

Figura 12. Ziegler Nichols lazo abierto. Adaptado de (Kushwah & Patra, 2014)

Para controladores con la forma  $K(s) = K_p + K_i/s + K_d s$ , las ganancias del controlador puede ser calculadas de la siguiente manera:

$$K_p = K \quad K_i = \frac{K}{T_i} \quad K_d = \frac{K}{T_d} \quad (5)$$

El **método de Cohen Coon** se utiliza para superar la respuesta lenta y constante que ocurre en Ziegler Nichols, usualmente utilizada en sistemas de primer orden (Normey-Rico, 2007). Basado en el mismo principio de la respuesta del sistema a un

cambio en la variable de ajuste (ver Figura 13), se tienen en cuenta los parámetros K (ganancia del sistema), T (constante de tiempo) y L (tiempo de retardo). Finalmente se remplaza en las reglas propuestas (ver Figura 14).

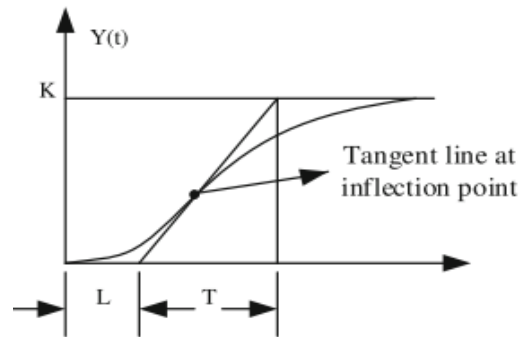


Figura 13. Curva reacción Cohen Coon. Tomado de (Badal et al., 2019).

Controller	Kc	Ti	Td
Standard PID	$\frac{1.35}{a} \left(1 + \frac{0.18T}{1-T}\right)$	$\frac{2.5 - 2T}{1 - 0.39T} L$	$\frac{0.37(1-T)}{1 - 0.81T} L$

Figura 14. Cohen Coon regla. Tomado de (Normey-Rico, 2007).

El **método Amigo** con el mismo principio de curva reacción busca calcular las constantes del controlador PID, las reglas del método propuesto son:

$$\begin{aligned}
 kp &= \frac{1}{k} \left(0.2 + 0.45 \frac{T}{L}\right) \\
 ki &= \left(\frac{0.4L + 0.8T}{L + 0.1T}\right) L \\
 kd &= \left(\frac{0.5LT}{0.3L + T}\right)
 \end{aligned}$$

Figura 15. Amigo regla. Tomado de (Hagglund, 1995)

Con el fin de determinar el controlador PID en eventos discretos y continuos, en este proyecto, se utiliza un proceso común basado en la **ingeniería dirigida por modelos**, este enfoque tiene como objetivo facilitar la construcción de software, a través de abstracciones y modelos reutilizables; también mejora la comunicación entre las partes interesadas y los desarrolladores. Los modelos nos ayudan a comprender un problema complejo y sus posibles soluciones a través de abstracciones. En la ingeniería dirigida por modelos, el enfoque principal y los productos son modelos más que programas de computadora (Selic, 2003). Como Selic menciona, la principal ventaja de la ingeniería dirigida por modelos es que los modelos pueden expresarse utilizando conceptos que están más cerca del dominio del problema en lugar de la tecnología informática.

Como ya sabemos la ingeniería dirigida por modelos radica en la reutilización de **modelos**, definidos como artefactos formulados en un lenguaje de modelado que describen la estructura y comportamiento del sistema (Kühne, 2004). La correcta definición de los modelos se logra a través de los **Metamodelos** los cuales permiten describir el modelo ofreciendo un grado de abstracción adecuado para que una herramienta de transformación pueda procesar los modelos, el metamodelo en sí es un modelo (Pascuas et al., 2015).

### **Capítulo 3**

#### **Estado del arte**

A continuación, se presenta un resumen acerca de la teoría de control, además se abarca enfoques importantes en relación con la ingeniería dirigida por modelos con sistemas de hardware y software desde el año 2002 hasta el presente año, destacando los siguientes referentes:

(Venkatesan, 2002) propone un método para controlar la calidad de productos a la salida de un proceso continuo, para ello se hacen pruebas de probabilidad para llevar la media de calidad del producto al punto de referencia o setpoint y brindar información si el proceso debe ser reajustado o no para alcanzar el objetivo. Como resultado de la investigación se discuten las ventajas y desventajas en aplicar una prueba de probabilidad para el ajuste de control.

(Chunjiang Qian & Wei Lin, 2002) mediante un método de diseño de dominación de retroalimentación, presentan un compensador de salida lineal que permite que el sistema en lazo cerrado sea exponencialmente estable, en presencia de una condición de crecimiento lineal. Como resultado se muestra cómo el compensador proporciona estabilidad simultánea en una familia de sistemas no lineales.

El trabajo propuesto por Leva (Leva et al., 2013) presenta una vasta información teórica sobre el uso de los principios básicos de un sistema de control con retroalimentación en sistemas de software, detallando los componentes básicos en este tipo de sistemas y algunas reglas de ajuste para calibrar un controlador PID, además de

algunos ejemplos específicos de aplicabilidad del proceso de retroalimentación en problemas de ingeniería de software.

Janert (2013) propone un enfoque prometedor que permite adaptar el control de retroalimentación en los sistemas informáticos, el enfoque está en una fase preliminar para enfrentar diferentes problemas de ingeniería de software, mediante una posible simulación del sistema, utilizando métodos heurísticos para el diseño del controlador, con el fin de obtener resultados eficientes. La investigación presenta la aplicación de la teoría en diferentes situaciones presentes en la ingeniería de software.

(Karlsson & Zhang, 2013) proponen un control de la retroalimentación en la industria de la publicidad donde ha demostrado ser tremendamente valioso, el objetivo de esta investigación es la implementación de un sistema capaz de cumplir con los objetivos establecidos por los anunciantes, para la toma de mejores decisiones a la hora de mostrar los anuncios. Como resultado se presenta la aplicación del control de publicidad en línea en una campaña, como un manual breve dirigido para ingenieros de control.

(Antonova & Batchkova, 2013) presentan un enfoque basado en la ingeniería dirigida por modelos para el desarrollo de sistema de control basado en el lenguaje UML y MARTE para el análisis del modelado, el enfoque muestra un ejemplo simple de control en un tanque de nivel, como resultado describe cómo la metodología combinada de UML y MARTE permite describir todo el ciclo de vida de un sistema de control, presentando el modelo de hardware y software del sistema.

La investigación de Chrouta (2015) propone sintetizar un controlador de retroalimentación para sistemas no lineales de tiempo discreto representados por modelos

difusos, el enfoque se basa en la minimización de un criterio cuadrático, el resultado de la investigación muestra que las variables del estado del sistema se comportaron satisfactoriamente.

(Arcelli & Cortellessa, 2016) detallan los desafíos presentes en la aplicación de la teoría de control en sistema de software, destacando en términos generales como ventaja principal cómo los controladores logran sus objetivos bajo ciertas circunstancias y como desventaja tener que modelar el sistema a controlar por ecuaciones. Como resultado de la investigación se describen todos los retos a abordar como la definición de unos objetivos claros, identificación de las posibles perturbaciones del sistema, la representación del sistema en ecuaciones, el diseño e integración del controlador y finalmente, el proceso de pruebas (testing) de un sistema de software con control.

Existen diversas aplicaciones del control con eventos variables como tecnologías relacionadas al internet de las cosas (IoT), por ejemplo (Kawamoto et al., 2017), propone un sistema con capacidad de control retroalimentado tanto para eventos continuos como discretos, con el fin de gestionar las multitudes presentes en centros comerciales, estadios entre otros. Como resultado se presenta la efectividad del enfoque en la gestión dinámica de multitudes bajo varios escenarios proporcionando resultados numéricos para la verificación del modelo.

(Auslander et al., 2017) describe cómo los computadores digitales utilizan los principios de control de retroalimentación en sistemas físicos, mostrando los beneficios del software en tiempo real en contraste con el software regular, se describe cómo se debe

realizar la implementación de sistemas de software con retroalimentación y las restricciones a tener en cuenta. Como resultado de la investigación se muestra la arquitectura del software con los métodos diseñados para una implementación genérica en diferentes sistemas, finalmente se presenta la implementación en el eje de un robot articulado para la validación del enfoque propuesto.

(Charfi et al., 2019) presenta un herramienta impulsada por la ingeniería dirigida por modelos, enfocada en los requisitos de sistema y la ingeniería de hardware, la arquitectura de la herramienta está centrada en tres niveles: nivel de fusión, nivel de arquitectura y nivel físico, además ofrece la posibilidad de generación automática de documentación para los requisitos y los modelos de hardware, como resultado se presenta el uso de la herramienta en el dominio automotriz y cómo contribuye en el cumplimiento de los requisitos.

(Jia et al., 2019) evalúa la implementación de un servidor web con virtudes de retroalimentación, para ello se lleva a cabo procesos experimentales, es decir verificando la salida del sistema en respuesta a una entrada, con el propósito de estimar una función de transferencia del sistema e implementar un controlador PI (proporcional e integral), los resultados de la investigación muestran como el sistema se estabiliza de acuerdo al valor de referencia establecido por el administrador del sitio web, siendo la variable a controlar en este caso el rendimiento del servidor.

(Magalhães et al., 2020) implementa un software que permite la automatización en plantas de producción, destacando dos características principales en la aplicación: identificación del modelo matemático de la planta utilizando el método de Sundaresan y



Krishnaswamy y estimación de un controlador P (proporcional), PI (proporcional e integral) y PID (proporcional, integral y derivativo), utilizando métodos heurísticos como Cohen Coon y Ziegler Nichols. El resultado de la investigación demuestra como el software mejora el rendimiento y confiabilidad del proceso de producción durante las operaciones de alta mar y tierra.

Las investigaciones presentes muestran una gran avance en la aplicación de sistemas retroalimentados en ciertos sectores, sin embargo, el enfoque utilizado sigue siendo demasiado específico, en el sentido que está centrado en la solución de un problema en particular, donde generalmente es necesario de un modelo matemático específico, por ende, no se propone una metodología o enfoque para aplicar el control a sistemas continuos y sistemas con eventos discretos de manera simultánea y generalizada.

La originalidad del enfoque de esta investigación radica en la aplicación de los principios de un sistema de control de retroalimentación en los sistemas de continuos y discretos mediante la ingeniería dirigida por modelos, la idea básica de la investigación es tomar la respuesta del sistema como entrada, mediante la recopilación de datos, para producir una estimación del sistema y diseño de un controlador PID, capaz de mantener las capacidades de retroalimentación como la estabilidad y el rendimiento, para ello se utiliza métodos experimentales o heurísticos , finalmente, con el diagrama de bloques se produce código fuente para la simulación del sistema continuo con un controlador PID y en el caso del sistema discreto se diseña el controlador que permita la estabilización según el comportamiento del sistema.

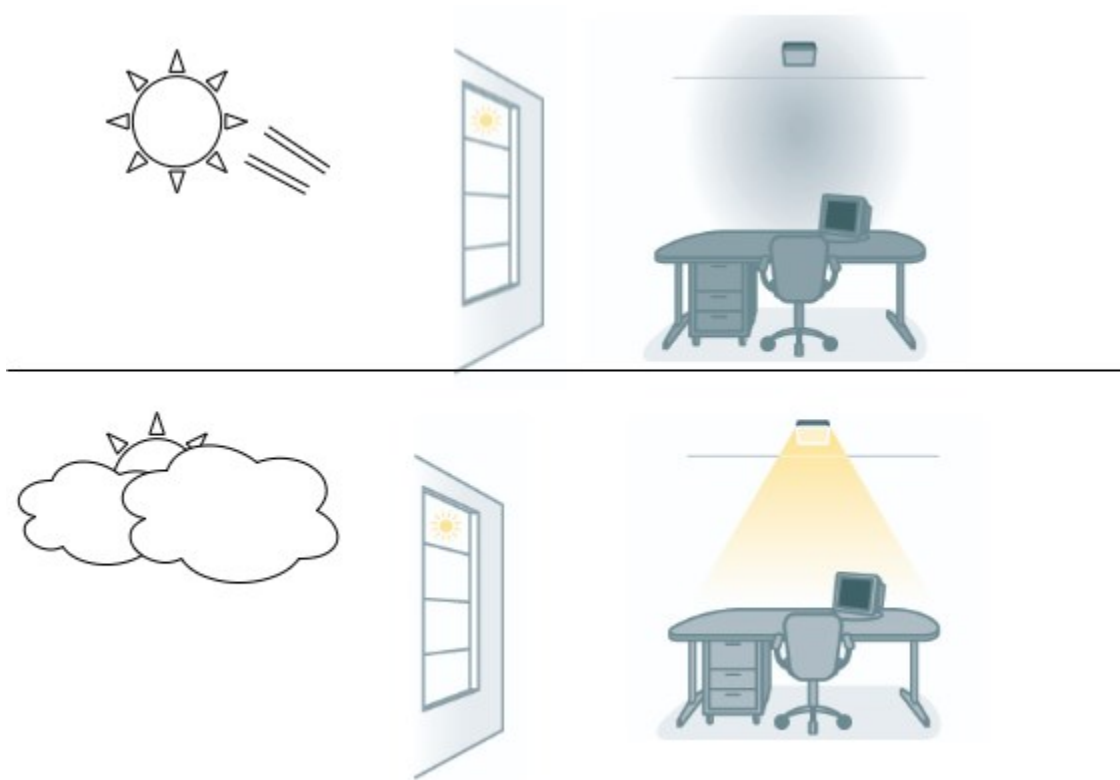
## Capítulo 4

### Ejemplos

A continuación, se detallan los ejemplos utilizados para la validación del enfoque tanto en eventos continuos como discretos. En primer lugar, se presentará un sistema continuo, posteriormente se describirá un sistema discreto.

#### 4.1 Sistema de iluminación

**4.1.1 Descripción del sistema.** El ejemplo comprende el sistema de control de iluminación, es un sistema común dentro de la industria, además es ampliamente utilizado en espacios residenciales, comerciales, etc. Este tipo de sistemas tienen como propósito mantener el nivel de iluminación deseado en un espacio específico según el contexto, es decir, si la luz natural incidente es alta, el nivel de iluminación deberá disminuir (Figura 16 parte superior), pero si la luz natural incidente es baja, el nivel de iluminación deberá aumentar (Figura 16 parte inferior). Para lograr este objetivo se pueden proponer dos soluciones: la primera consiste en encender la luminaria a la hora en que comúnmente empieza a oscurecer y apagarla cuando se empiece a percibir mayor luz natural. Esta solución puede acarrear dificultades ya que las horas de mayor cantidad de luz natural incidente pueden variar, y en ocasiones, durante la misma jornada se pueden presentar muchas variaciones en el nivel de luz natural.



*Figura 16. Sistema de iluminación.*

Otra solución, más efectiva, consiste en instalar un dispositivo (fotocelda, fototransistor, etc.) para detectar la cantidad de luz natural incidente, y de acuerdo con esto, encender o apagar la luminaria de forma automática. En esta situación, la cantidad deseada de luz será comparada con la cantidad de luz natural incidente, lo que ocasionará que se accione o no el interruptor de la luminaria. Para este caso es necesario el uso de un controlador que sea capaz de mantener el nivel de iluminación, reduciendo los efectos percibidos o perturbaciones presentes en el sistema.

**4.1.2 Arquitectura e implementación del sistema.** La Figura 17 muestra el diseño del sistema de control de iluminación, a manera de prototipo, comprende un diodo emisor de luz o LED de potencia y un resistor dependiente de luz o LDR, el cual detecta el nivel de iluminación emitido por el LED, cabe recordar que para que este prototipo sea funcional, el resistor debe estar a una distancia no mayor a 15 cm del LED, con el fin de que el resistor pueda detectar la luz del led de potencia y no la luz del ambiente.

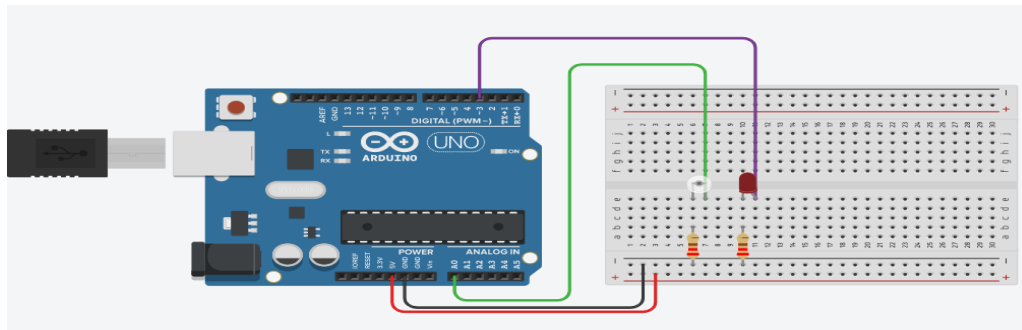


Figura 17. Diseño electrónico sistema de iluminación.

Este sistema está programado en un microcontrolador (Arduino Uno). La Figura 18 muestra la conexión eléctrica en detalle.

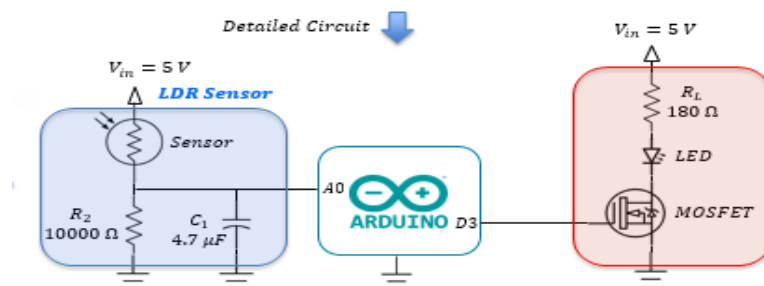


Figura 18. Circuito electrónico sistema de iluminación.

Para este ejemplo se toma como base el código disponible en <https://github.com/andres4588/Sistema-de-iluminacion> y en el Anexo 1. Con el fin de verificar el comportamiento de la respuesta del sistema, a una entrada de una función unitaria o a un cambio en el setpoint, la curva de respuesta debe tomar la forma de una función sigmoidea para poder aplicar las reglas de ajuste de un controlador PID, como se puede observar en la Figura 19.

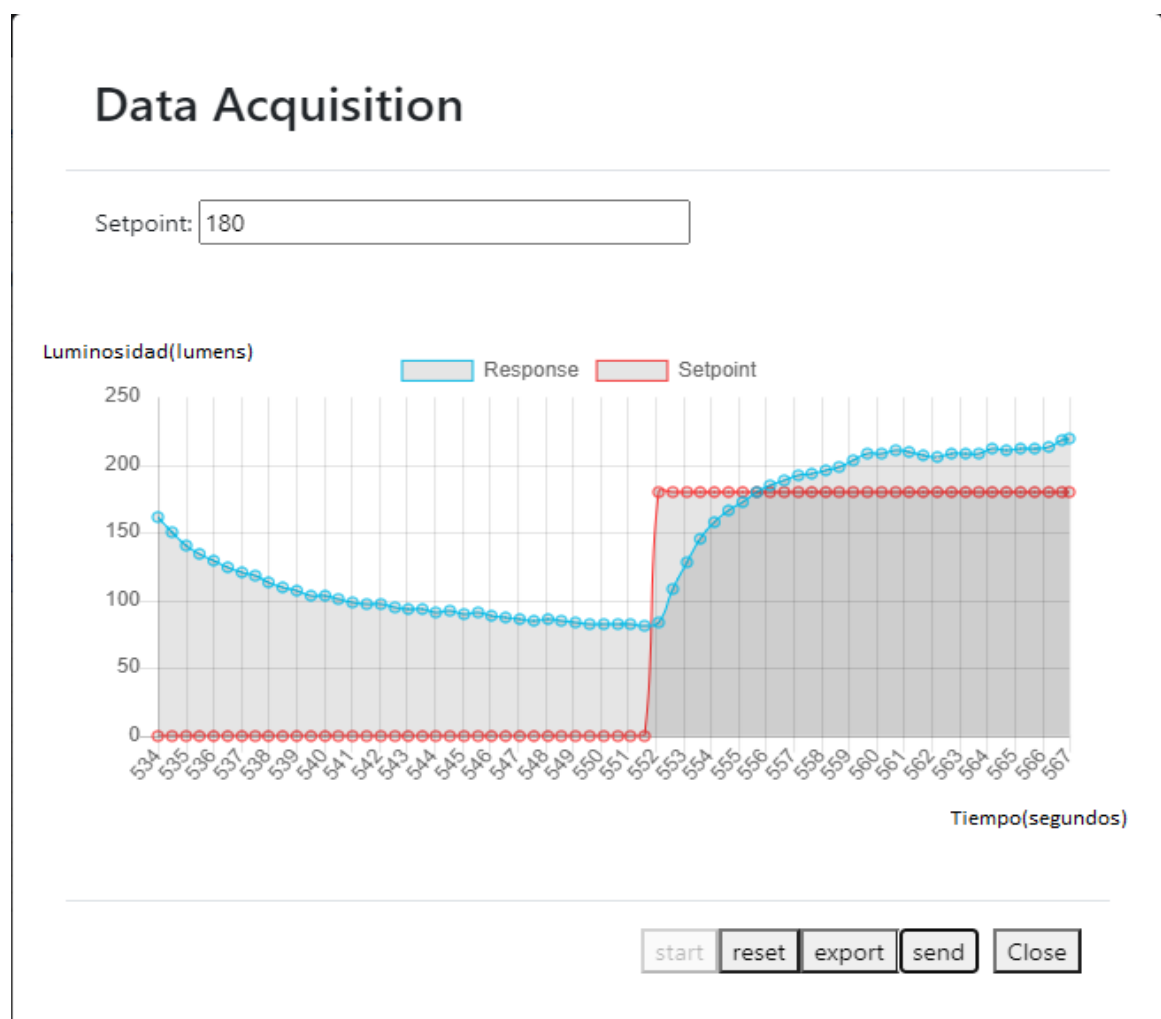


Figura 19. Adquisición de datos sistema iluminación.

## 4.2 Sistema de venta virtual de prendas de vestir

**4.2.1 Descripción del sistema.** El proceso de ventas en una tienda virtual de prendas de vestir comprende un sistema común dentro de la dinámica discreta, para ello se desarrolló una aplicación donde el usuario o cliente puede autenticarse en el sistema, ver los productos disponibles con su respectiva descripción y precio (ver Figura 20), además se añadió la opción de un carrito de compra para guardar los artículos a comprar, finalmente cuando el usuario desea aceptar la compra se muestra el nombre del usuario y la cantidad de los artículos seleccionados y el monto a pagar por ellos (ver Figura 21), mientras que también se tiene un perfil de administrador, el cual puede gestionar los productos, ver el total de productos vendidos y los ingresos generados. El código fuente del sistema está disponible en <https://github.com/andres4588/Shopping>.

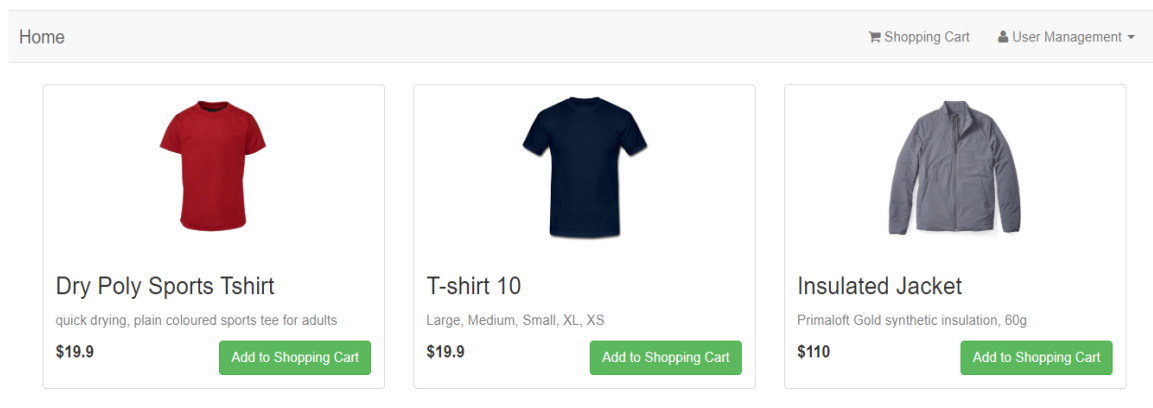


Figura 20. Menú inicial tienda ropa.



*Figura 21. Menú lista de compras.*

Las particularidades de este proceso demuestran las similitudes deseables en sistemas con mecanismos de retroalimentación:

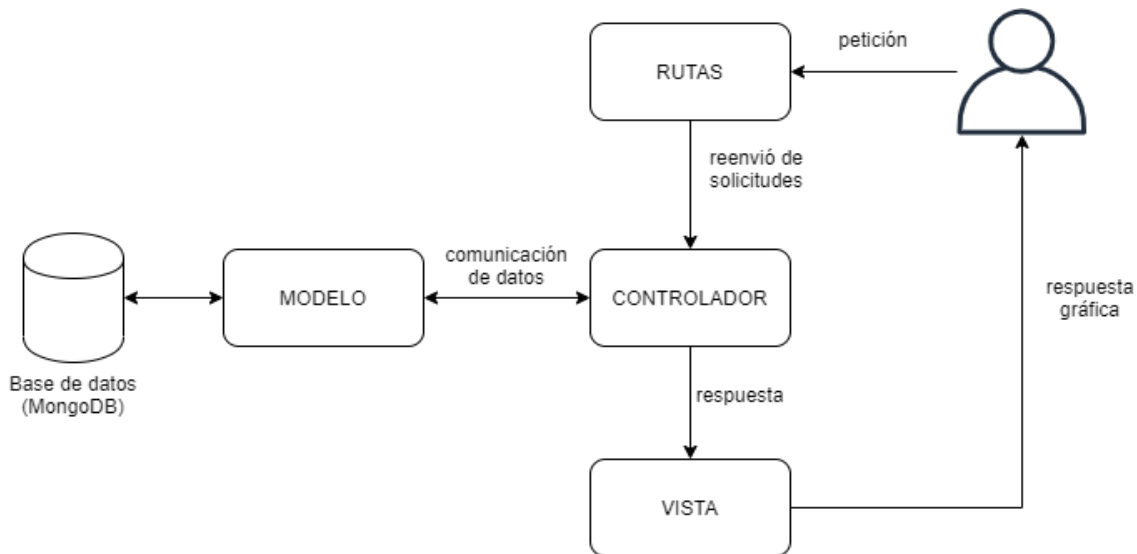
- El modelo analítico del proceso es desconocido o simplemente incompleto.
- El proceso cambia constantemente (debido a la oferta y la demanda u otros factores).
- Tiene un objetivo claro, que es la cantidad de ventas necesarias para alcanzar el punto de equilibrio en relación con sus ingresos.

Específicamente, el proceso que consideramos es una tienda que necesita controlar el número de ventas realizadas, en el cual podemos alterar el porcentaje de descuento del precio de un producto para variar el número de ventas, lo que se sabe del sistema es que un mayor porcentaje de descuento tiende a generar un mayor número de ventas.

**4.2.2 Arquitectura e implementación del sistema.** El desarrollo de la aplicación está basado en el patrón de arquitectura modelo vista controlador (MVC) (Figura 22), con

el fin de brindar simplicidad y mantenibilidad al sistema, la ejecución de sistema está centrada en los siguientes pasos:

- El usuario realiza una petición al navegador mediante una ruta (dirección electrónica de acceso)
- El controlador recibe la solicitud y le solicita al modelo la información.
- El modelo responde al controlador con la información solicitada.
- El controlador envía los datos a la vista para organizar y aplicar estilo a la información.
- La vista proporciona una respuesta gráfica de la información solicitada.



*Figura 22. Arquitectura sistema de ventas.*

Como es usual en la ingeniería de control, el primer paso para abordar un sistema de control automático es diseñar un modelo del proceso a ser controlado. Dicho modelo representa el comportamiento del sistema a controlar y es de gran utilidad para analizar y



evaluar subsistemas o el sistema como tal. Para el proceso de adquisición de datos del sistema en un tiempo continuo se aplica un cambio escalón, en este tipo de sistemas con eventos discretos se hace un cambio en la variable de ajuste (porcentaje de descuento en el precio), para nuestro ejemplo verificaremos el número de ventas cuando en la tienda no hay un porcentaje de descuento en el precio y cuando se emplea un porcentaje de descuento en los productos.

El comportamiento de este tipo de sistemas puede variar según los requisitos, condiciones externas e internas, oferta-demanda, por ende, se decide desarrollar una simulación de un proceso de ventas con distintos comportamientos: un comportamiento donde el número de ventas es proporcional al porcentaje de descuento en el precio (ver Figura 23 parte superior) y un comportamiento donde el número de ventas representa una escala logarítmica en relación al porcentaje de descuento en el precio (ver Figura 23 parte inferior).

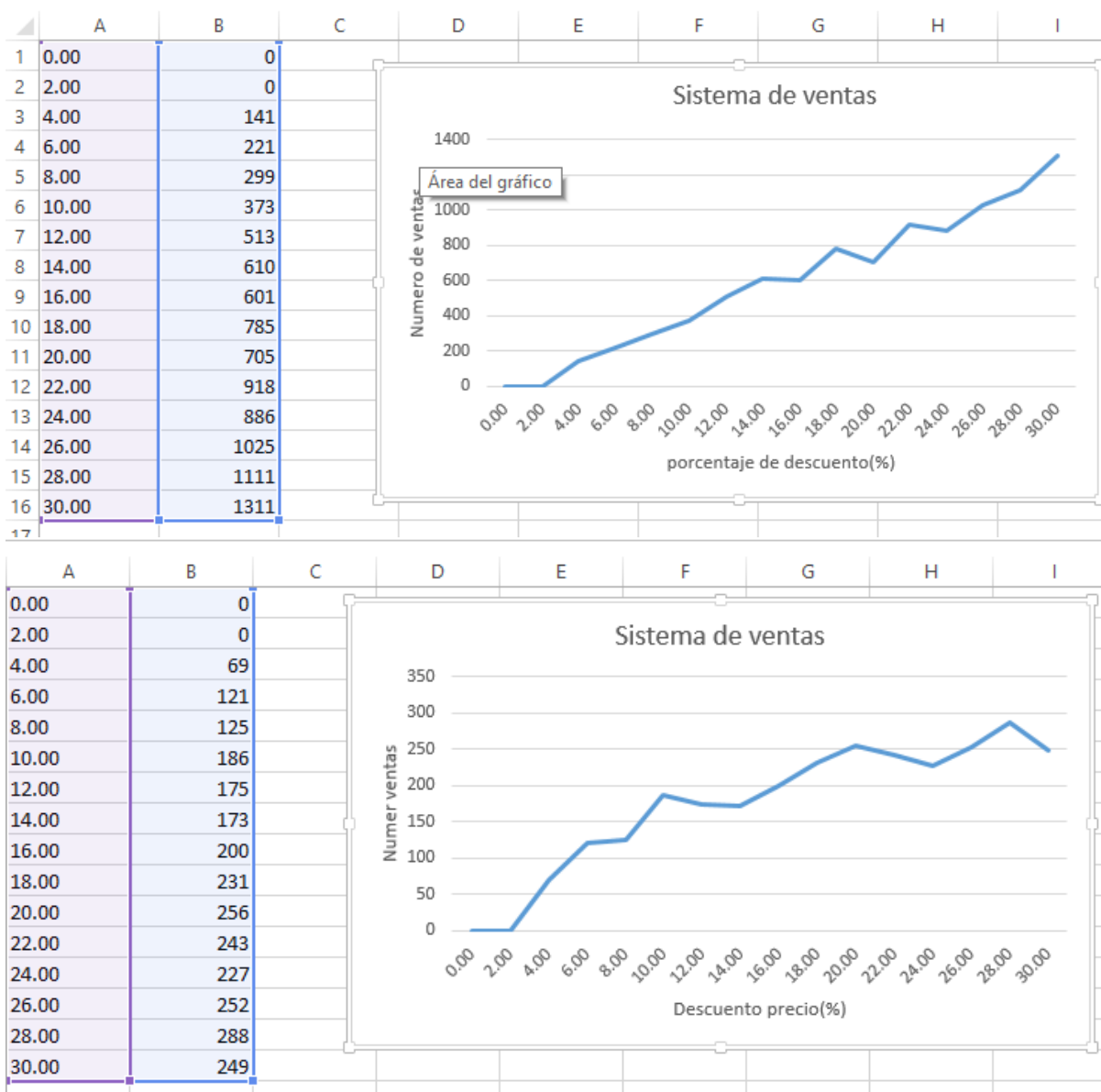


Figura 23. Número de ventas vs Porcentaje de descuento.

## Capítulo 5

### Resumen de la propuesta

Este enfoque se desarrolla principalmente para mitigar problemas en aquellos sistemas donde la información recibida cambia constantemente y es necesario actuar sobre dicha información para tomar las mejores decisiones. El desarrollo del enfoque propuesto está estructurado en dos grandes etapas, una etapa preliminar, donde se evalúa el sistema sin un controlador y una etapa de control del sistema con un controlador PID como se puede observar en la Figura 24. Los elementos representados en la parte superior corresponden al nombre de las subetapas, mientras que los elementos de la parte media corresponden a los resultados obtenidos en la subetapa correspondiente. La salida de una subetapa ingresa como entrada para la subetapa siguiente. En la última subetapa el usuario puede generar código Arduino para la simulación de un sistema control continuo, mientras que para los sistemas con eventos discretos se estima las ganancias del controlador PID de acuerdo con el comportamiento (historial de los datos) del sistema.

El enfoque propuesto toma como entrada: el comportamiento del sistema sin control, es decir la dinámica del sistema a una respuesta de una función unitaria o a un cambio de setpoint y produce como salida: las ganancias calculadas del controlador, se busca proporcionar los parámetros del controlador PID de diferentes reglas heurísticas.

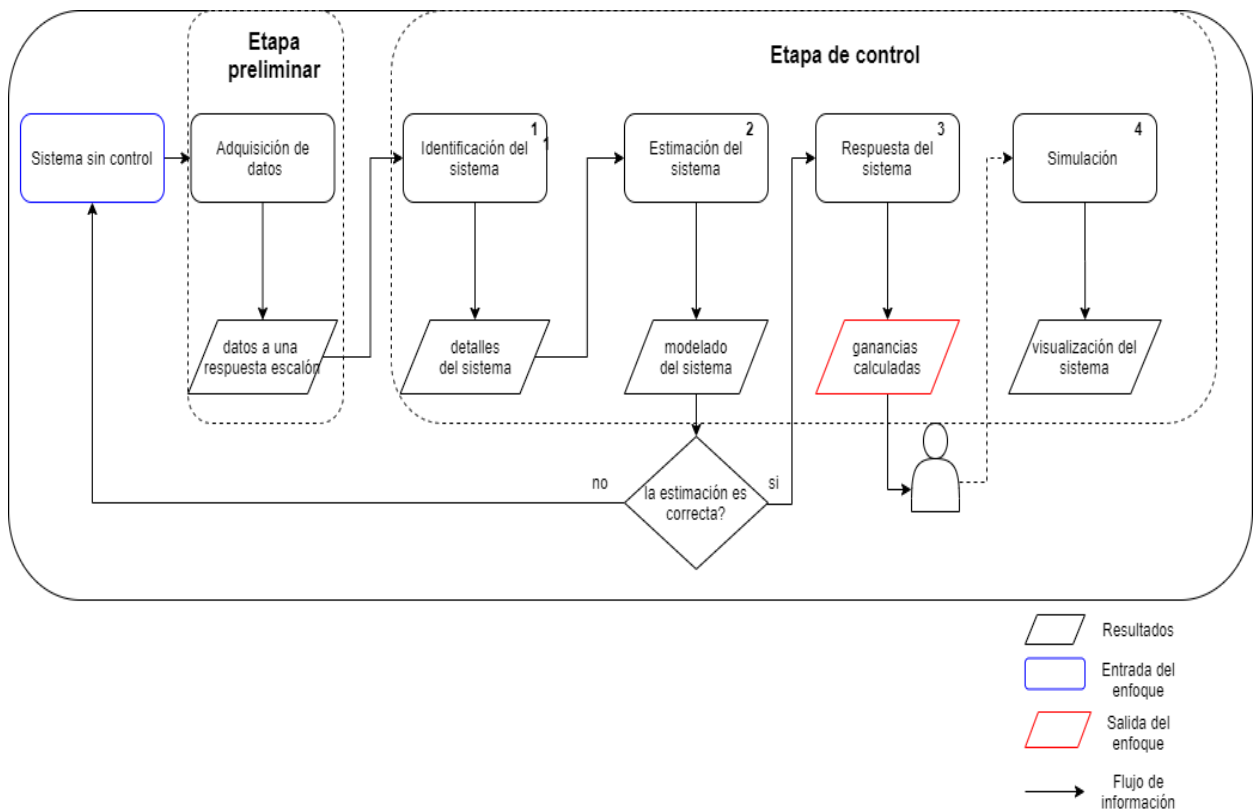


Figura 24. Enfoque control retroalimentado.

La originalidad del enfoque propuesto radica en la aplicación de los principios de un sistema de control de retroalimentación en sistemas de continuos y sistemas con eventos discretos, el proceso para el control de un sistema es conocido, el propósito de este trabajo es partir de ese proceso y adaptarlo realizar una implementación que permita su aplicación tanto en sistemas continuos como en sistemas discretos, donde en la primera etapa, denominada etapa preliminar se recogen datos del sistema variando la variable de ajuste o setpoint hasta que el sistema alcanza la estabilidad o el estado estable.

En la segunda etapa, denominada etapa de control, se lleva a cabo un proceso de ajuste de la curva para determinar el modelo analítico de los datos recogidos, determinando mediante métodos heurísticos las constantes del controlador PID. Además, de producir código fuente Arduino para sistemas continuos mediante la ingeniería dirigida por modelos. En las siguientes secciones se explica con más detalle cada uno de los componentes del enfoque, y las directrices para utilizar esos componentes conjuntamente.

## **Capítulo 6**

### **Propuesta**

El enfoque permite el proceso de adaptación de un sistema sin control a un sistema con control, para ello se presentan en detalle las etapas y subetapas propuestas, así como su implementación dentro de la plataforma VariaMos, finalmente, se presenta la aplicación en los dos casos de estudio, uno para sistemas con eventos continuos (sistema de iluminación) y otro para sistemas con eventos discretos (sistema de ventas), dentro del marco de VariaMos.

#### **6.1. Etapa preliminar**

En esta etapa se realiza la adquisición de datos del sistema, para ello el usuario aplica un cambio en la variable de ajuste (entrada del sistema) para obtener la dinámica del sistema (salida del sistema), la respuesta del sistema debe tender a un punto de estabilidad para poder aplicar reglas heurísticas durante el diseño del controlador, por ejemplo, para el sistema de discreto el usuario puede cambiar el porcentaje del precio del sistema que es la variable de ajuste para obtener el número de ventas que corresponde a la salida del sistema.

#### **6.2 Etapa de control**

Con la dinámica del sistema obtenida en la etapa preliminar se procede a calcular el modelo analítico del sistema y las ganancias del controlador PID, con el fin de evaluar

la respuesta del sistema con un controlador. Al utilizar un proceso experimental para llevar a cabo esta fase más no el proceso matemático, esta etapa suele requerir de un proceso iterativo durante la subetapa de estimación del sistema para obtener mayor exactitud (ver Figura 24).

En el diseño de los sistemas de control se determinan la configuración, las especificaciones y la identificación de las variables más representativas del sistema propuesto con el fin de mantener el comportamiento del sistema. Esta etapa está segmentada en las siguientes sub-etapas:

1. Identificación del sistema: El usuario ingresa parámetros como la variable de ajuste utilizada en la etapa preliminar, si la dinámica del sistema presenta retraso y si el sistema presenta dinámica continua o discreta, esta subetapa produce como resultado los detalles descritos anteriormente. Un conocimiento preciso del comportamiento del sistema permite estimar la respuesta del sistema y seleccionar la acción de control adecuado para mejorarla.
2. Estimación del sistema: Con los parámetros del sistema ingresados en la etapa anterior se presenta la aproximación de la respuesta del sistema a una dinámica de un sistema de primer o segundo orden subamortiguado, calculando la función de transferencia del sistema, presentando la transformada de Laplace para sistemas con eventos continuos y la transformada de Z para sistemas con eventos discretos. Esta subetapa produce como resultado las ganancias del controlador utilizando los métodos heurísticos desarrollados por Ziegler Nichols, Cohen Coon y Amigo.

3. Respuesta del sistema: Con las ganancias obtenidas en la subetapa anterior, se verifica la dinámica del sistema con el controlador, las metodologías heurísticas no siempre proporcionan un grado de precisión o estabilidad necesario, por ende, es posible ajustar manualmente las ganancias si es necesario, con el fin de producir como resultado las ganancias del controlador ajustadas.
4. Simulación del sistema: Finalmente, en esta subetapa el usuario puede volver a simular el sistema, si se trata de un sistema continuo en la plataforma de VariaMos, con las ganancias obtenidas en la subetapa anterior para evaluar el comportamiento del sistema; mientras que en sistemas con eventos discretos se proporciona los parámetros del controlador para ser simulados en un sistema externo, con el fin de verificar que el sistema cumpla con las condiciones preestablecidas, por ejemplo, que el sistema alcance el punto de referencia deseado, el error debe disminuir en un estado estacionario o que el rendimiento del sistema sea el adecuado de acuerdo a las posibles limitaciones del sistema.

### 6.3 Implementación

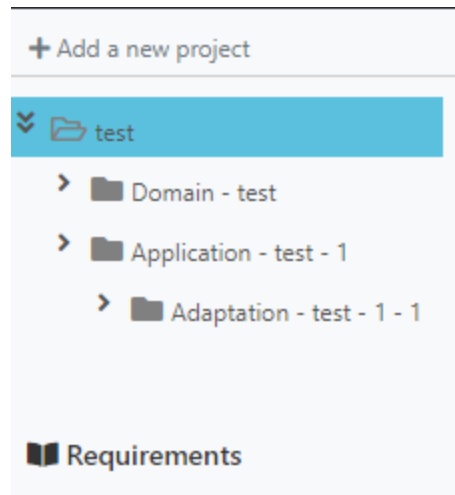
El enfoque propuesto se implementó como un módulo para la plataforma VariaMos<sup>3</sup>, el módulo se denomina Control y permite aplicar control retroalimentado a sistemas discretos y continuos. Para crear un proyecto de control en VariaMos, es necesario crear

---

<sup>3</sup> <http://variamos.dis.eafit.edu.co/>



un nuevo proyecto, esta acción genera un árbol de trabajo (ver Figura 25) que incluye los modelos de control en la carpeta Aplicación (Application).



*Figura 25. Árbol de trabajo VariaMos.*

La Figura 25 presenta el módulo de control en VariaMos, en la parte derecha se encuentra la paleta (Palette) de elementos para el diseño del diagrama de bloques de un sistema de control. La parte central corresponde a la zona de trabajo donde se modela un sistema de control con los elementos disponibles en la paleta. El menú de aplicación (Application Menu) permite la generación de código y la conexión para la visualización de datos en tiempo real de sistemas implementados en Arduino.

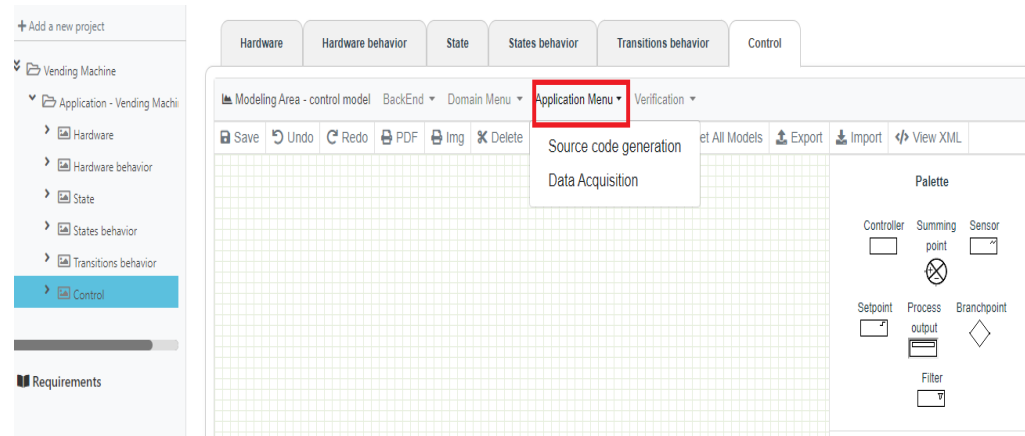


Figura 26. Menú control.

**6.3.1 Metamodelo.** Los conceptos principales de un sistema de control de retroalimentación se basan en estudios previos sobre ingeniería de control (Astrom & Wittenmark, 2011). Como parte de este trabajo se desarrolló un metamodelo para representar los componentes principales del control de retroalimentación como se muestra en la Figura 27. El metamodelo está constituido por 4 componentes principales:

- Bloque (Block): Tiene un nombre o descripción particular, un bloque puede ser de alguno de los siguientes tipos:
  - Referencia (Setpoint): Representa el valor de referencia del sistema, este elemento tiene 3 propiedades: **SetPoint** variable que representa el valor de referencia deseado, **time** variable que representa tiempo para la simulación de la respuesta del sistema con un controlador, y **Tm** que permite representar el número de ciclos de retroalimentación deseados para la simulación de la respuesta del sistema con un controlador.

- Controlador (Controller): Representa el controlador PID del sistema, tiene tres variables de valores positivos: **proportional** que representa la ganancia proporcional, **integral** que representa la ganancia integral, **derivative** que representa la ganancia derivativa.
- Filtro (Filter): Representa el filtro de datos en un sistema mediante la variable **filterType** que representa los tipos de filtros disponibles: **average**, el cual utiliza un algoritmo para el filtro de señales mediante el promedio de los datos y **recursive** que utiliza una regla exponencial.
- Sensor: Representa el sensor del sistema, la variable **initialPosition** representa el valor de posición inicial del sensor, cuyo valor por defecto es 0.
- Salida del sistema (Process Output): Representa la salida del sistema mediante la variable **currentOutput**, la cual indica el valor de salida inicial de un sistema, cuyo valor por defecto es 0.
- Planta (Plant): Este bloque representa el sistema a controlar.
- Relación (Relation): Representan el flujo de información del sistema de control retroalimentado.
- Punto de suma (SummingPoint): Representa la operación aritmética a utilizar entre el valor de referencia y la salida del sistema, mediante la variable **direction** que permite seleccionar diferentes operaciones.
- Punto de ramificación (BranchPoint): Elemento que representa la bifurcación de una señal a diferentes bloques.

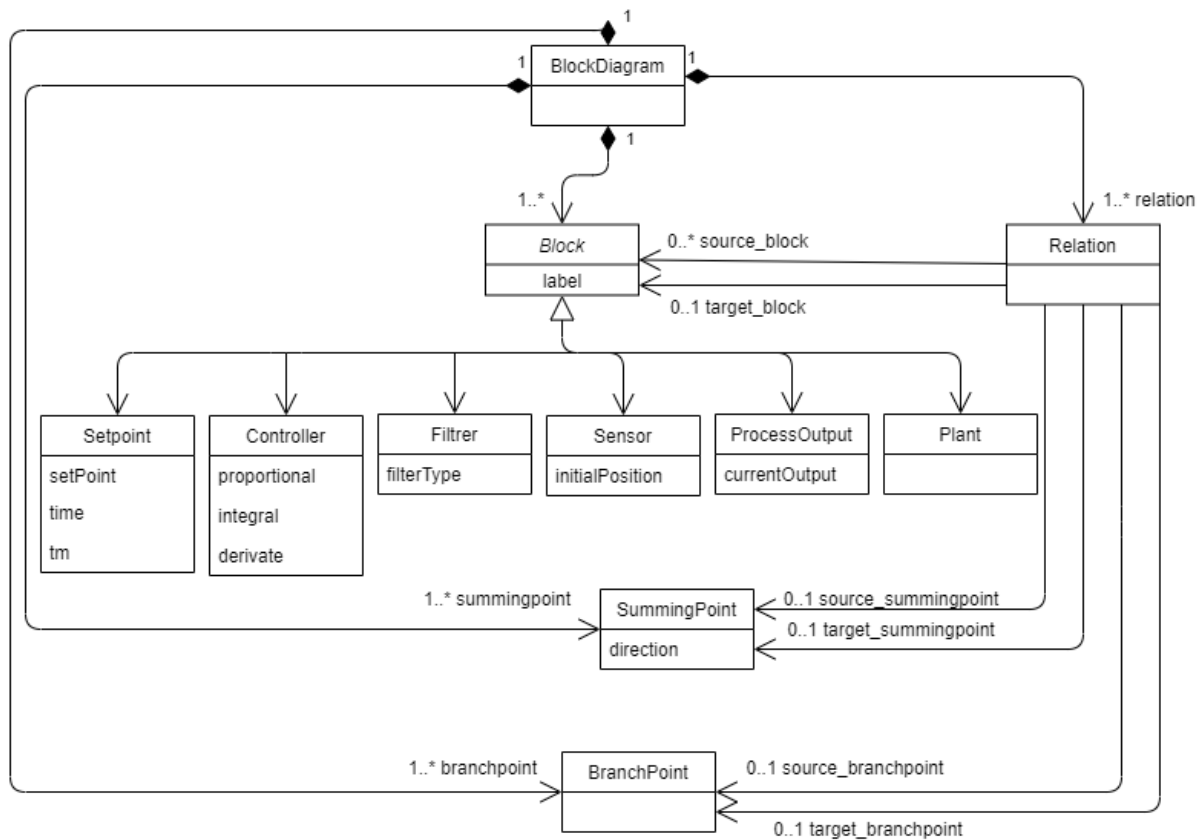


Figura 27. Metamodelo sistema de control.

Se describe una serie de restricciones detalladas del metamodelo de un sistema de control:

- Un bloque sólo puede tener una relación de entrada y una relación de salida.
- Sólo puede haber un bloque tipo setpoint especificado en el modelo.
- Sólo puede haber un bloque tipo salida del sistema.
- Las relaciones deben ser unidireccionales.
- El bloque de salida del sistema no debe tener relaciones de salida.
- El punto de suma sólo tiene una relación de salida.

**6.3.2 Modelo.** La sintaxis concreta conforme al metamodelo definido anteriormente se presenta en la Figura 28, la Figura 29, la Figura 30 y la Figura 31:

#### 6.3.2.1 Bloques.







Nombre	Notación grafica	Descripción
Referencia (Setpoint)		Representa el valor que se desea alcanzar.
Controlador (Controller)		Representa el controlador PID.
Filtro (Filter)		Representa el filtro que se aplica a una señal.
Sensor		Representa el valor actual o medido del sistema.
Salida del sistema (Process output)		Representa la salida del sistema.
Planta (Plant)		Representa el sistema que se va a controlar.

Figura 28. Bloques sistema de control.

#### 6.3.2.2 Relaciones.

Relación		Representa la dirección del flujo de información.
----------	---	---

Figura 29. Relaciones sistema de control.

#### 6.3.2.3 Punto de suma.


Punto de suma		Produce una suma algebraica de las entradas
---------------	---	---

Figura 30. Punto de suma.

#### 6.3.2.4 Punto de ramificación.


Punto de ramificación		Representa la bifurcación de una señal
-----------------------	---	--

Figura 31. Punto de ramificación.

**6.3.3 Visión general.** La Figura 32 muestra el enfoque en un sistema continuo, a través de las siguientes etapas:

1. El sistema sin controlador se comunica con el servidor (Variamos Server) para poder publicar datos en VariaMos Web en tiempo real.
2. Cuando se están publicando los datos, el usuario puede exportar la información necesaria para el proceso de adquisición de datos (descrito en el Capítulo 4).
3. Los datos generados en el paso anterior son la entrada necesaria para el proceso de control (Capítulo 6), con el fin de obtener el modelo matemático del sistema y las ganancias del controlador PID.
4. Una vez obtenidos los parámetros del controlador podemos generar el código del comportamiento del sistema con el controlador diseñado.
5. Finalmente, igual que en la etapa 1, podemos publicar los datos del sistema con controlador para ver el comportamiento del sistema en tiempo real.

## Visión general sistemas continuos

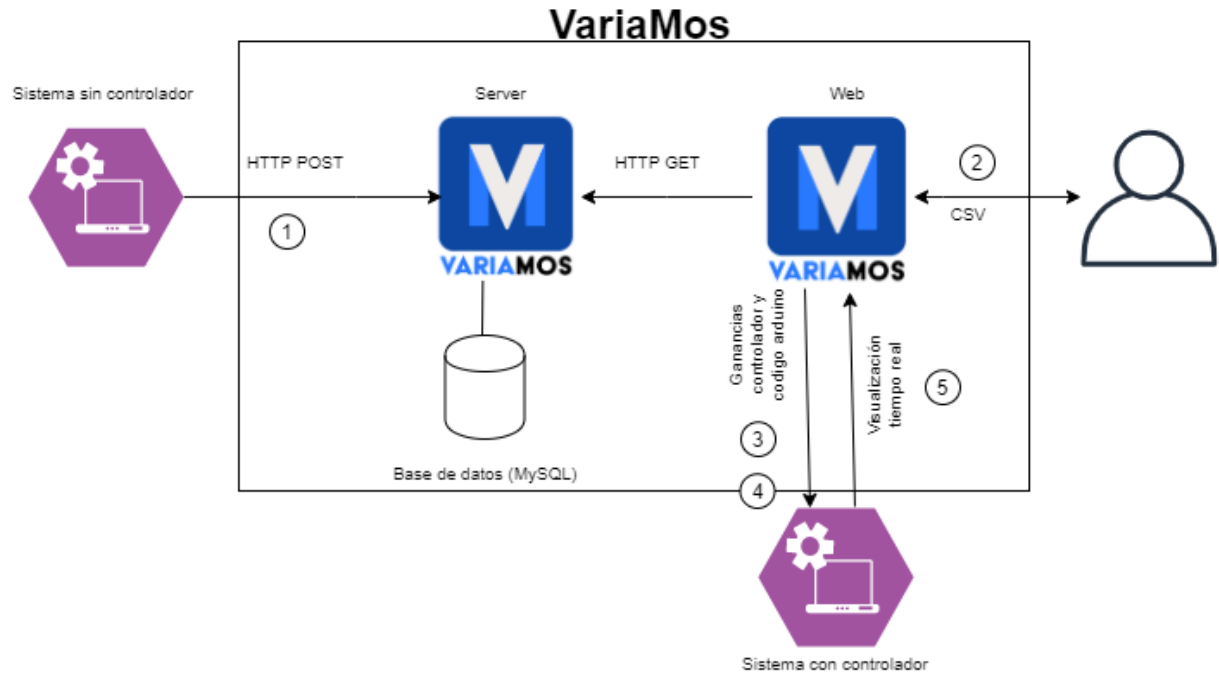


Figura 32. Visión general en un sistema continuo.

La Figura 33 muestra el enfoque en un sistema discreto segmentado en las siguientes etapas:

1. Capturar los datos del sistema sin controlador comúnmente es el historial.
2. Los datos obtenidos en el paso anterior son la entrada necesaria para el proceso de control, con el fin de obtener el modelo matemático del sistema y las ganancias del controlador PID.
3. Una vez obtenidos los parámetros del controlador podemos verificar la dinámica del sistema con el controlador diseñado.

## Visión general sistemas discretos

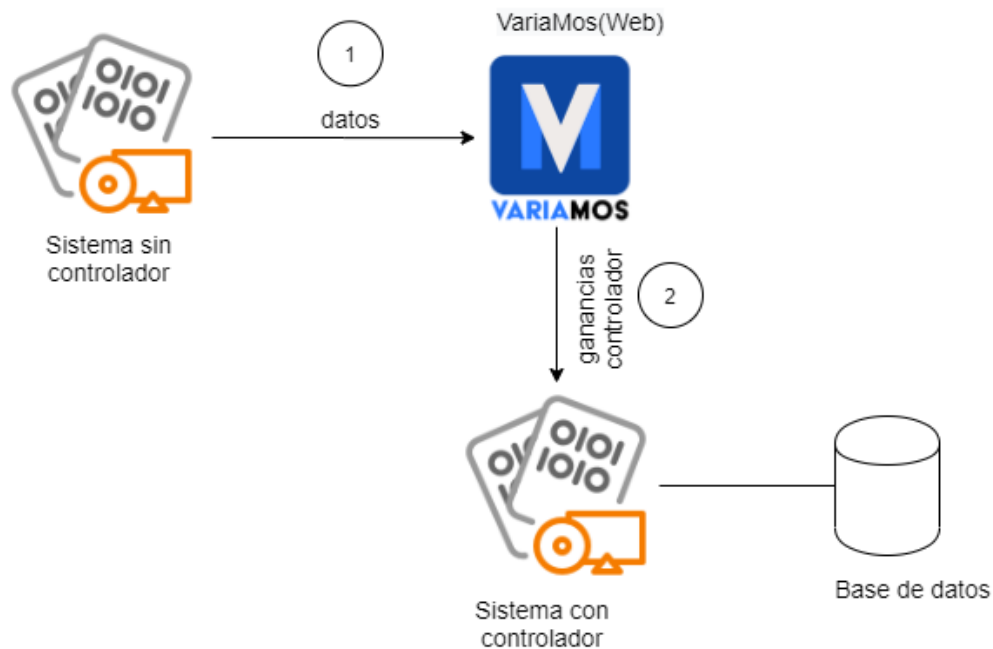


Figura 33. Visión general en un sistema discreto.

El proceso de control para ambos tipos de sistemas, de manera muy general, está segmentado en dos grandes etapas:

1. Comienza con un sistema sin controlador, la herramienta de software VariaMos generamos el archivo CSV del comportamiento a una entrada escalón.
2. Termina produciendo el modelo del sistema y las ganancias del controlador PID diseñado, para visualizar el resultado del sistema con un controlador.

### 6.4 Aplicación en sistema continuo

Para el uso de este enfoque, se espera recibir como entrada un sistema sin control (cuyo modelado no se considera en nuestro alcance). Para este ejemplo, el modelado de



hardware y de comportamiento del sistema de realizó con el módulo de modelado de hardware de VariaMos. La Figura 35 presenta el modelado del sistema, cabe anotar que en esta investigación nos centramos en la acción de control (bordeada en color rojo).

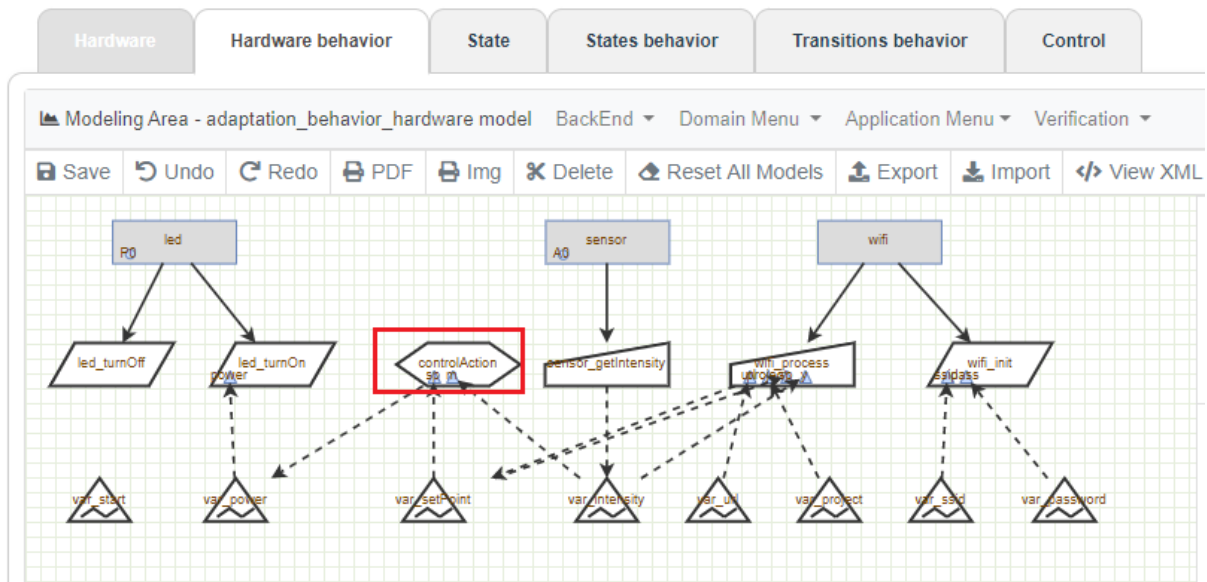


Figura 34. Comportamiento sistema de iluminación.

**6.4.1 Identificación del sistema.** La Figura 36 presenta el esquema del sistema de iluminación descrito en el Capítulo 4, para realizar esta actividad se usó el módulo de control implementado en la plataforma VariaMos, al modelar un elemento **Control Action** en módulo de hardware y de comportamiento se crea automáticamente un elemento tipo **Control Action** en el módulo de control el cual corresponde al actuador en un sistema de control, en nuestro caso es un led de potencia.

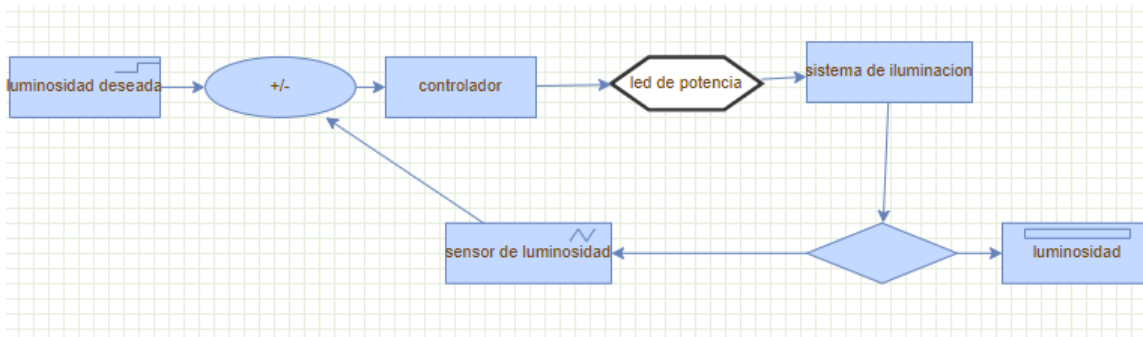


Figura 35. Diagrama del bloque sistema de iluminación.

**6.4.1.1 Setpoint.** Para nuestro ejemplo, el setpoint se definirá de acuerdo con el brillo deseado. La Figura 36 ilustra la configuración definida.

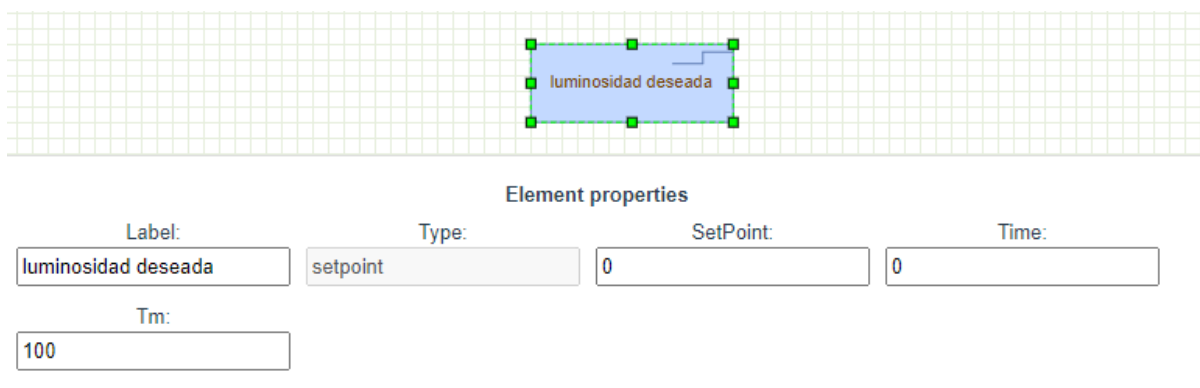


Figura 36. Setpoint sistema de iluminación.

**6.4.1.2 Controlador.** Las constantes del controlador pueden ser ajustadas manualmente, además se obtendrán en la fase de estimación del modelo (Figura 37).

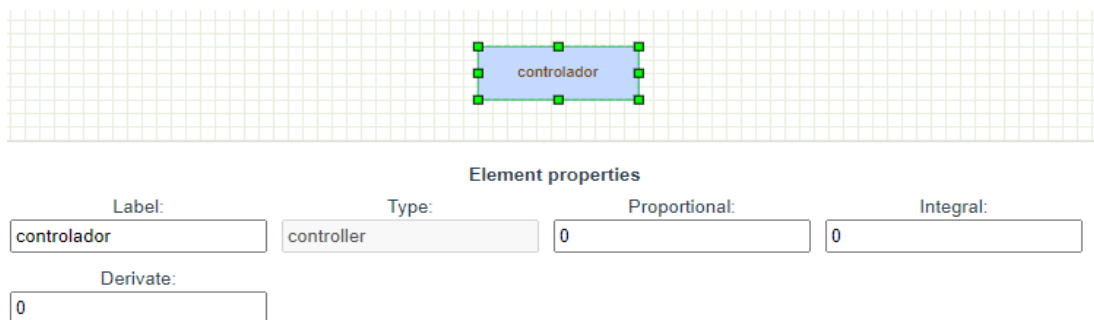


Figura 37. Controlador sistema de iluminación.

**6.4.1.3 Punto de suma.** Seleccionaremos la operación a ejecutar entre el setpoint y el comportamiento actual del sistema de iluminación (Figura 38).

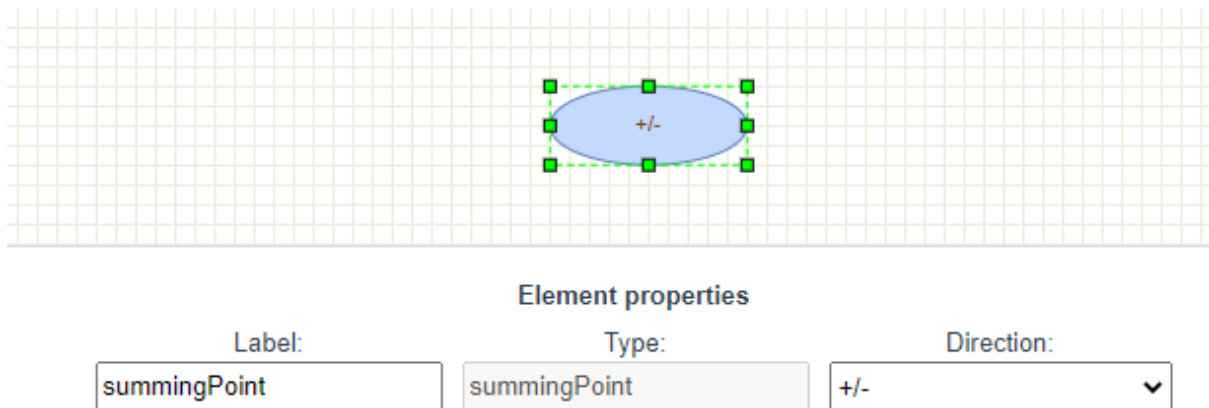


Figura 38. Punto de suma sistema de iluminación.

**6.4.1.4 Sensor.** Con este elemento obtendremos el estado actual del sistema de iluminación (Figura 39).

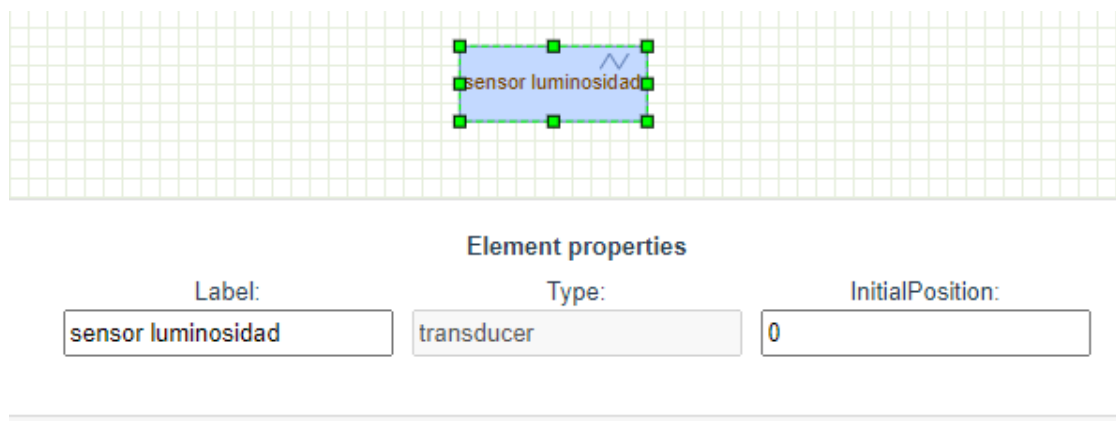


Figura 39. Sensor sistema de iluminación.

**6.4.1.5 Salida del sistema.** Determinaremos la salida del sistema que en nuestro caso será la iluminación del led de potencia (Figura 40).

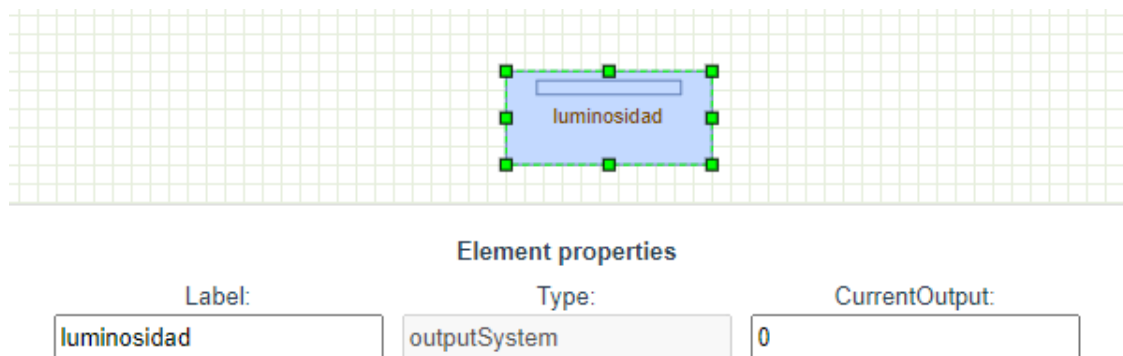


Figura 40. Salida del sistema de iluminación.

**6.4.1.6 Acción de control.** Detallaremos el actuador que se desea controlar sistema en este caso es un led de potencia (Figura 41).

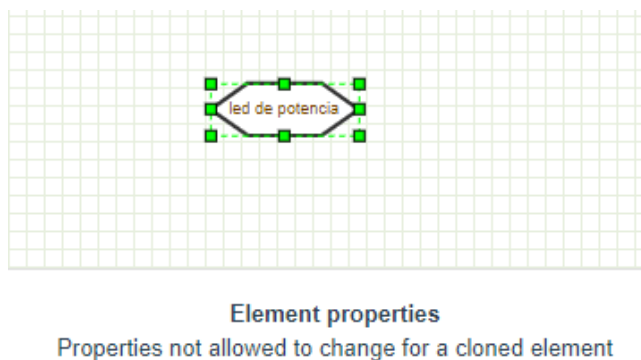


Figura 41. Acción de control Sistema de iluminación.

**6.4.1.7 Planta.** Representa el sistema a controlar (Figura 42).

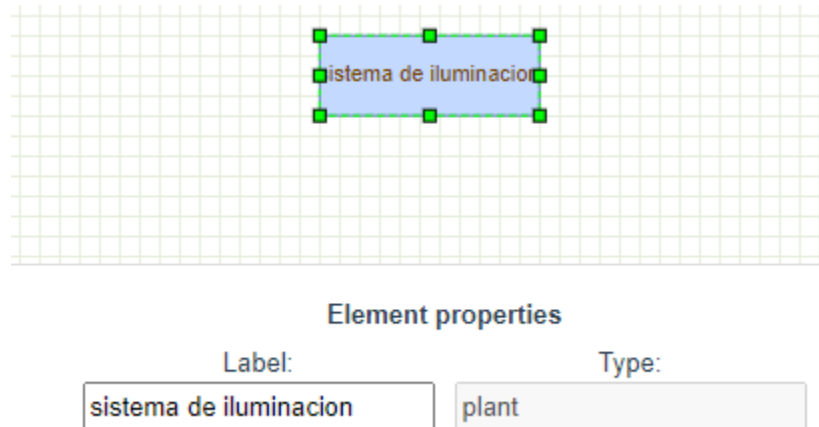


Figura 42. Planta sistema de iluminación.

Para identificar el modelo del sistema se debe detallar el rango de ajuste aplicado en el escalón unitario o el setpoint utilizado en la fase de evaluación del sistema sin un controlador (Capítulo 4), determinaremos si el sistema presenta retardo con respecto al tiempo y si la dinámica del sistema es continua o discreta, finalmente importar los datos obtenidos en la etapa anterior de adquisición de datos.

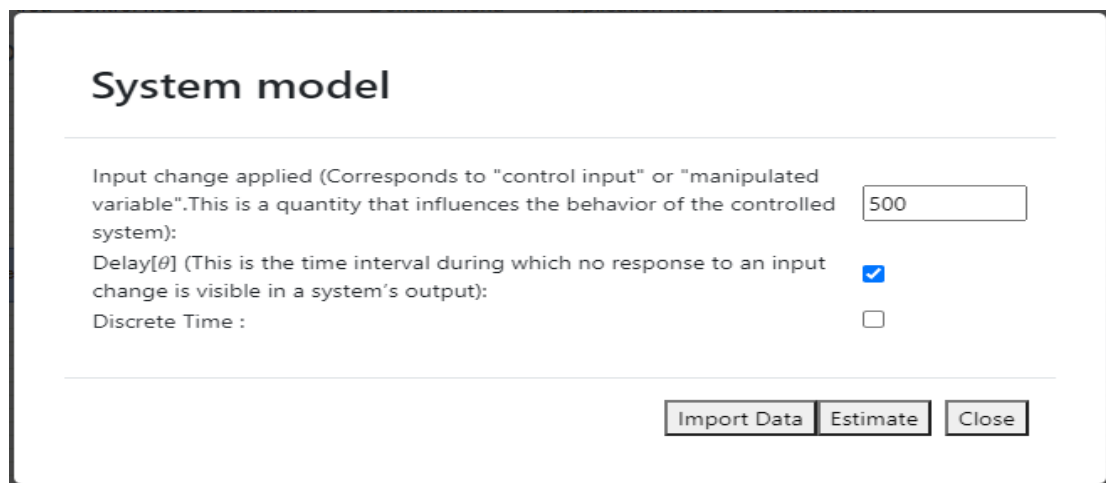


Figura 43. Identificación sistema de iluminación.

**6.4.2 Estimación del sistema.** En esta etapa se estima la salida del sistema utilizando regresión no lineal, con el fin de estimar la salida del sistema a una función de primer orden:

- Tiempo de retardo: 0.5 segundos, equivalente al tiempo que se demora en responder el sistema al cambio en la entrada del sistema.
- Constante de tiempo (T): 0.76 segundos, esta variable corresponde al tiempo que se demora el sistema en llegar al 63% del valor final (Simmonds-Mendoza et al., 2018).
- Ganancia del sistema:  $K = \frac{\Delta OUT}{\Delta IN} = \frac{445-0}{500-0} = 0.89$

Donde  $\Delta OUT$  representa la diferencia entre rango de salida del sistema en nuestro ejemplo es de 445 (Figura 44 línea azul) y  $\Delta IN$  a la entrada aplicada (ver Figura 43).

Finalmente, se estima las ganancias del controlador utilizando las reglas desarrolladas por Ziegler-Nichols, Cohen Coon y Amigo para el ajuste del controlador PID.

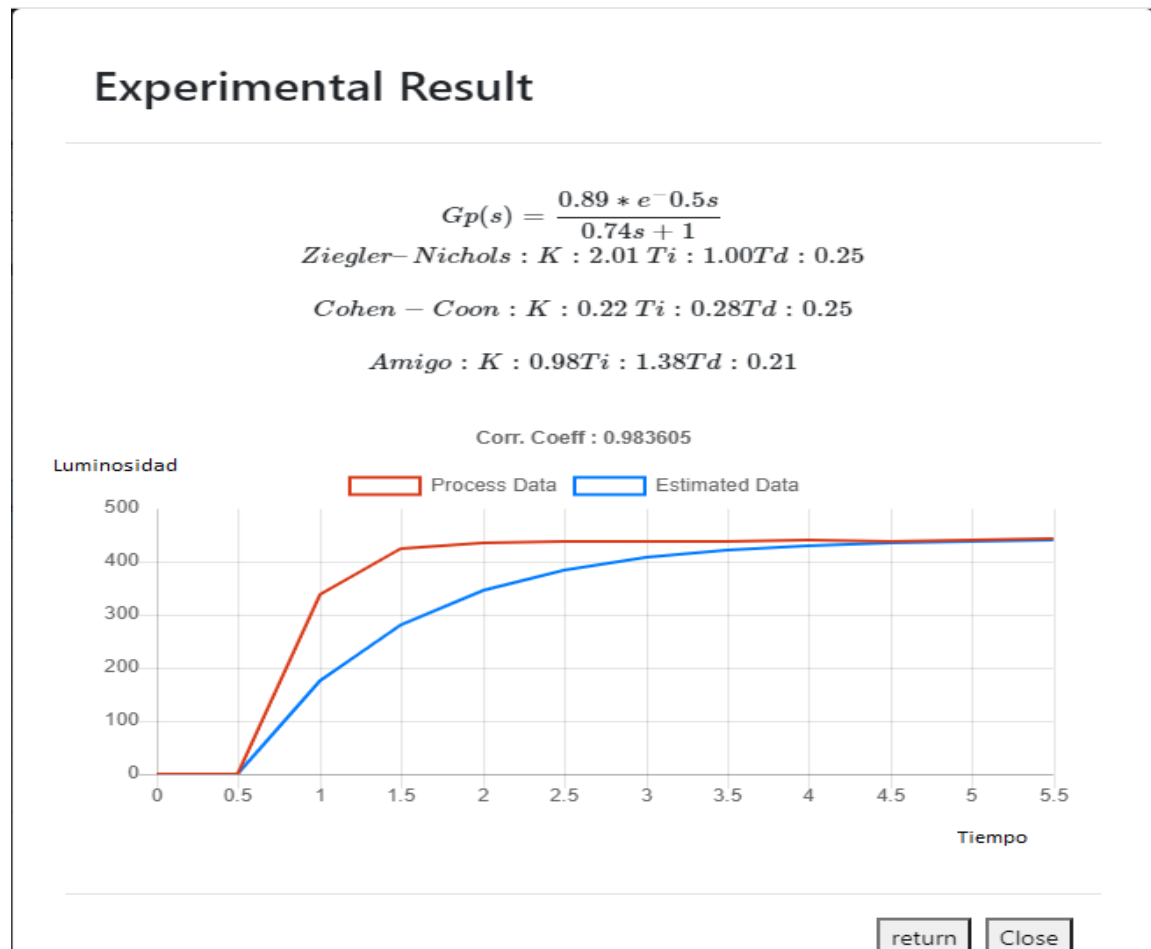


Figura 44. Estimación sistema de iluminación.

**6.4.3 Respuesta del sistema con controlador.** Verificaremos la respuesta del sistema con el controlador. En la Figura 45 podemos observar que hay un ligero sobreimpulso aproximadamente desde los 7 segundos, es decir cuando la salida del sistema sobrepasa el punto de referencia, cabe mencionar que las metodologías heurísticas no siempre proporcionan un desempeño o estabilidad en el sistema como métodos analíticos, por lo tanto es posible ajustar las ganancias manualmente si es necesario.

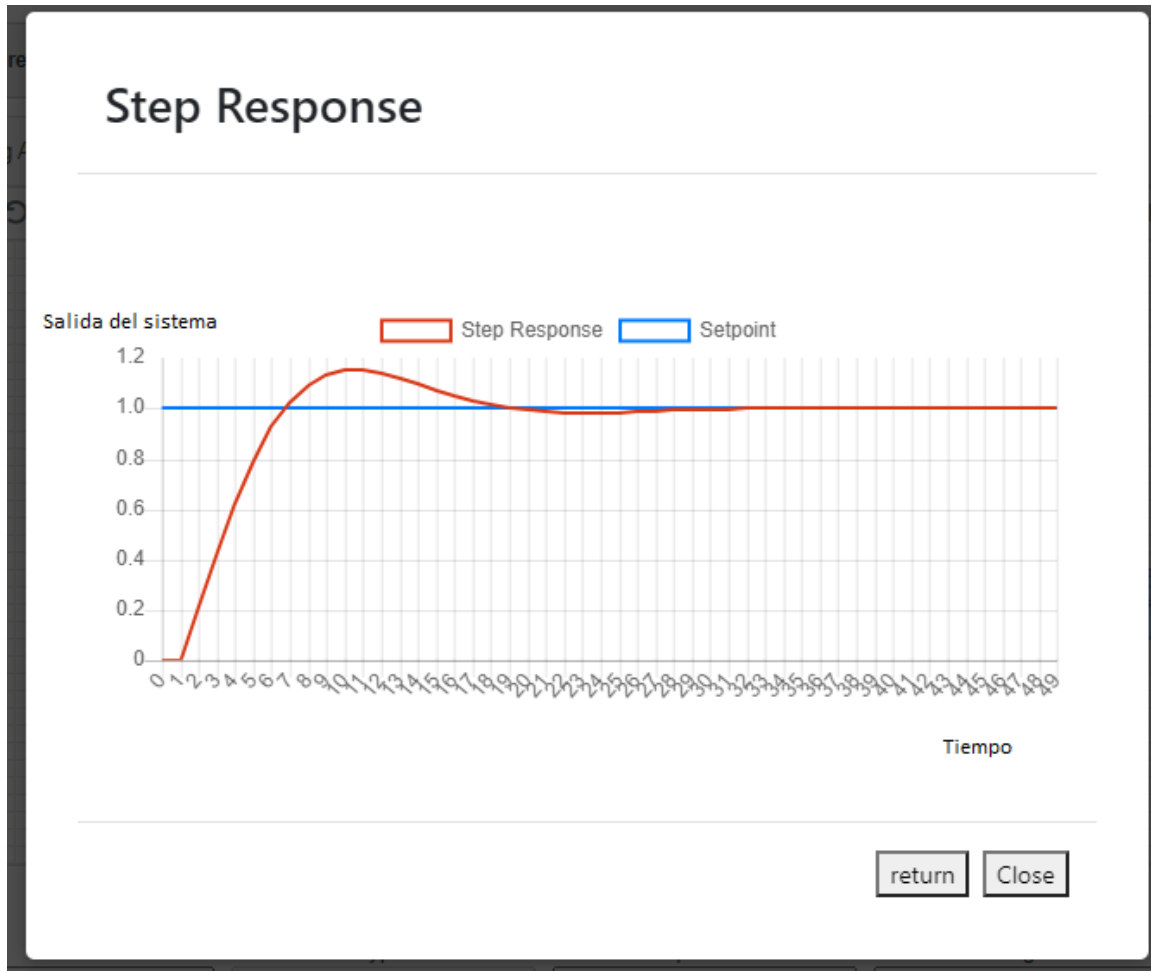


Figura 45. Salida sistema de iluminación con control.

**6.4.4 Simulación del sistema de iluminación.** Para finalizar el ejemplo verificaremos el comportamiento del sistema del código generado con el controlador PID diseñado (<https://github.com/andres4588/Sistema-de-iluminacion>). Como se puede observar en la Figura 46 la respuesta del sistema (iluminación) con un controlador PID tiende a estabilizarse aproximadamente a los 4 segundos al setpoint ingresado por el usuario (120).



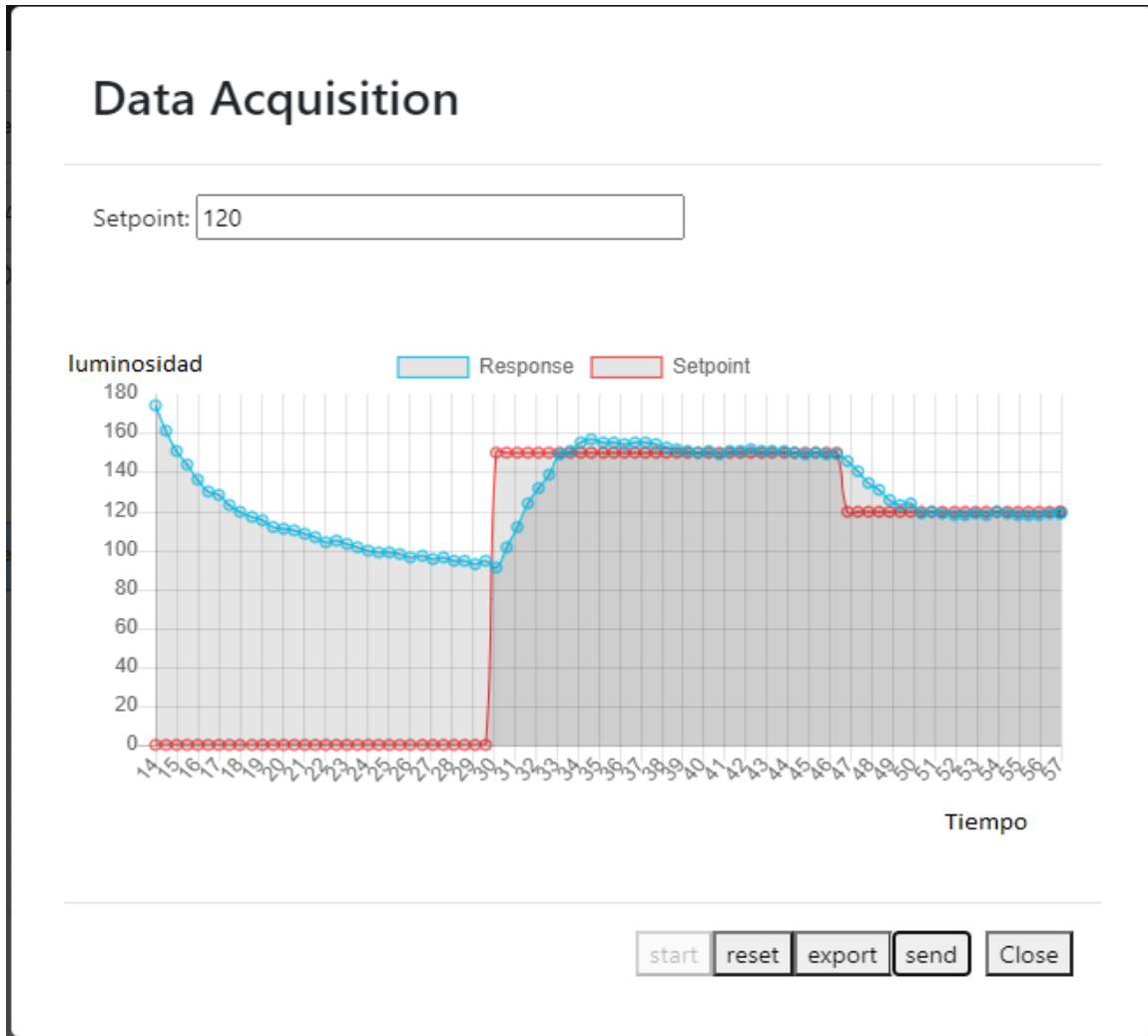


Figura 46. Simulación del sistema de iluminación con control.

### 6.5 Aplicación en sistema discreto

Como que se describió en el Capítulo 4, en este sistema se desea controlar el número de ventas y se tiene una entrada que es el porcentaje de descuento, al variar este porcentaje del precio podemos modificar el número de ventas, la complejidad en este sistema es la toma de datos ya que por ejemplo, el proceso de retroalimentación se hace cada día es decir, solo podremos ajustar el precio una vez se tengan el número

de ventas diarias, si deseamos ver la respuesta del sistema (número de ventas) a un cambio en el porcentaje precio el sistema puede tardar varios días o incluso meses en llegar a un punto de estabilidad, por ende, se decide hacer una simulación de la respuesta del sistema con el controlador PID.

Podemos modelar el sistema usando un diagrama de bloques (ver Figura 47), donde se tiene un setpoint que es el número de ventas deseadas, el controlador del sistema, tenemos la planta corresponde al sistema de ventas y la salida es las ventas actuales

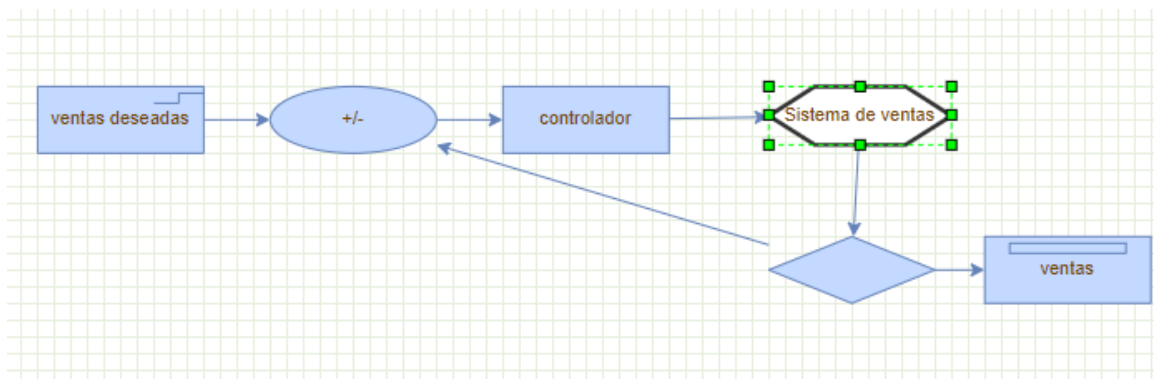


Figura 47. Sistema de ventas.

**6.5.1 Identificación del sistema.** En este sistema con eventos discretos, evaluaremos el cambio aplicado a nuestra variable de entrada para simular los datos de la salida del sistema; para nuestro caso de estudio el cambio aplicado fue del 30% de descuento en el precio de los productos (ver descripción detallada en el Capítulo 4) y seleccionaremos que el sistema pertenece a eventos discretos (Figura 48).

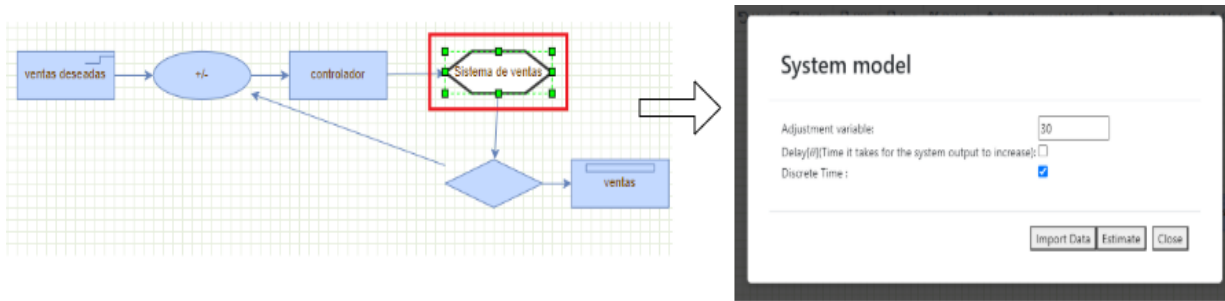


Figura 48. Identificación del sistema de ventas.

**6.5.2 Estimación del sistema.** Aunque el proceso presenta varios picos en varios instantes de tiempo el sistema tiende a estabilizarse aproximadamente a las 140 ventas (línea color rojo), por ende, podemos estimar la salida del sistema de ventas con una salida de un sistema de primer orden o segundo orden (línea color azul), para discretizar el sistema y estimar un modelo matemático en términos de la transformada Z, igualmente se muestra las ganancias con los métodos heurísticos experimentales para el diseño del controlador (ver Figura 49).

## Experimental Result

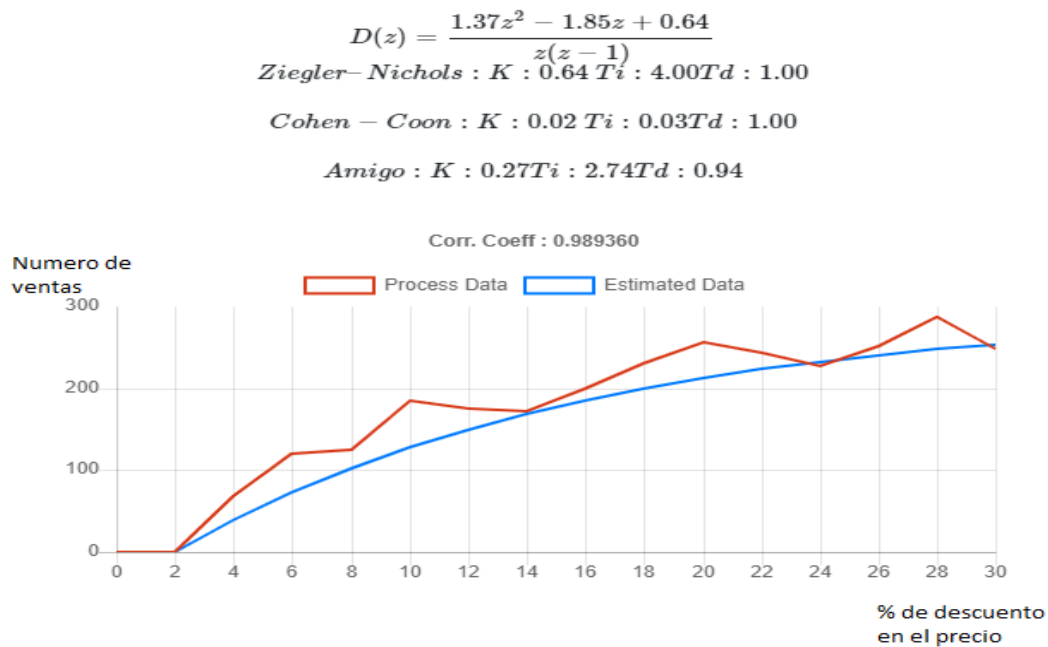


Figura 49. Estimación del sistema de ventas.

**6.5.3 Identificación del sistema.** Con las ganancias obtenidas en el paso anterior verificaremos que la respuesta del controlador sea la adecuada para alcanzar un punto de estabilidad (ver Figura 50), recordando que la herramienta de VariaMos permite hacer cambios manuales de las ganancias del controlador para una mejor estimación del controlador PID.

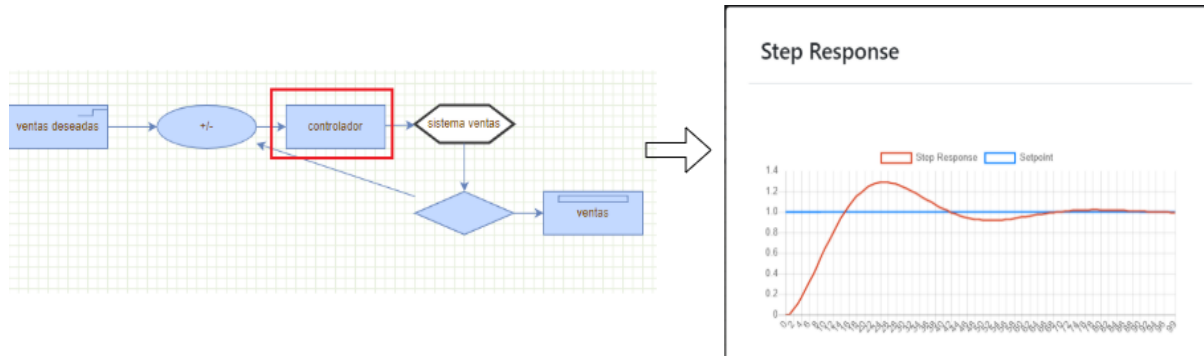


Figura 50. Respuesta del controlador en el sistema de ventas.

**6.5.4 Simulación del sistema.** Se presenta la simulación del sistema de ventas con un controlador para ello se establece primero un setpoint de 250 ventas y para verificar el comportamiento del sistema a un cambio en la referencia durante el proceso de ventas, se decide cambiar el setpoint a 150 ventas, en la Figura 51 podemos observar como el sistema de ventas se ajusta con el fin de alcanzar los dos valores de referencia ingresados.

Además, se puede observar como el sistema presenta una mejor precisión cuando el valor de referencia es más pequeño, en el caso de 250 ventas como valor de referencia el sistema tiende a oscilar en 30 ventas con respecto al setpoint, si el valor de referencia es de 150 ventas el sistema tiende a oscilar entre 20 ventas.

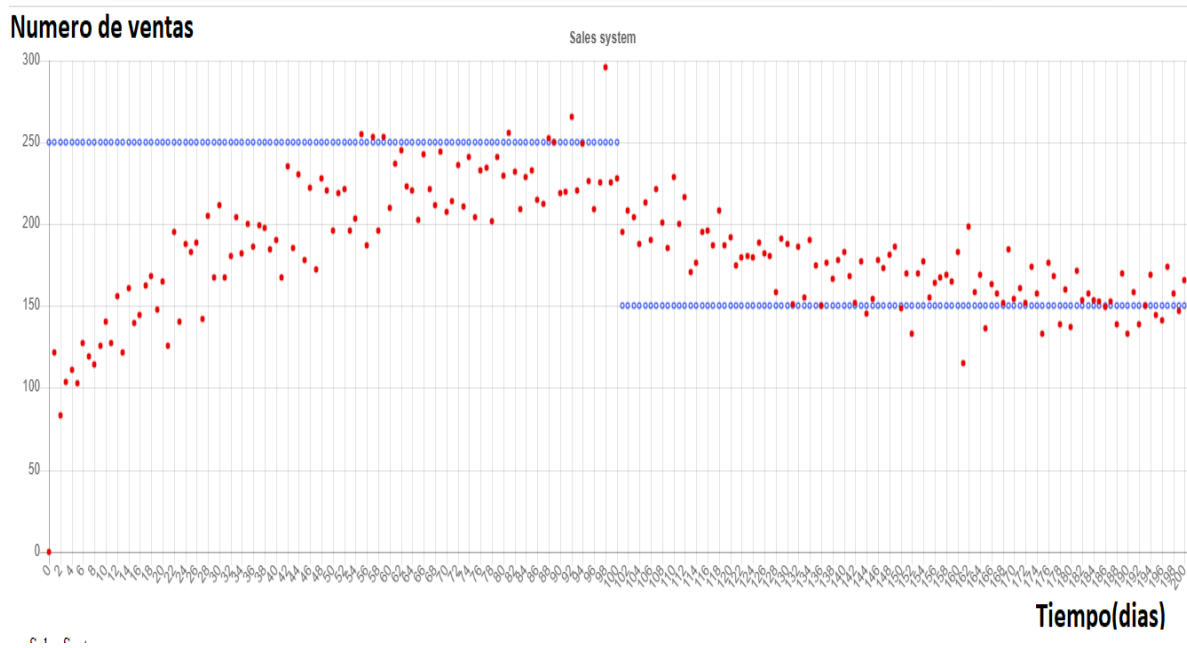


Figura 51 Simulación sistema de ventas con control.

## Capítulo 7

### Evaluación de la propuesta

Los requisitos básicos que debe satisfacer un sistema de control retroalimentado se centran en las tres características que se determinaron en el proceso de validación del enfoque propuesto en este documento.

- **Estabilidad:** Determina si el comportamiento de un sistema es estable o no, al aplicar acciones correctivas demasiado grandes los sistemas tienden a oscilar de forma permanente provocando la inestabilidad del mismo. Un sistema es estable cuando la salida permanece en el mismo estado. Un sistema de control inestable puede producir oscilaciones persistentes o de gran amplitud en la señal. Desde el punto de vista del control, una respuesta inestable es indeseada, existen varios métodos que permite verificar la estabilidad del sistema como Root Locus, matriz de Routh, mapa de polos y ceros, respuesta temporal, entre otros. Ya que este es un enfoque más experimental utilizaremos la respuesta del sistema a un cambio en la entrada para evaluar la estabilidad del sistema.
- **Desempeño:** Necesitamos que nuestro sistema llegue al punto deseado lo más rápido posible. Un sistema de control debe completar su respuesta a una señal de entrada en un tiempo aceptable.
- **Precisión:** Es la capacidad del sistema de dar el mismo resultado en instantes diferentes, estando sometido a las mismas condiciones. Los sistemas necesitan ser lo suficientemente precisos para evitar problemas altamente críticos. Un sistema de control debe ser capaz de reducir cualquier error a un límite aceptable.

En el caso del ejemplo del sistema continuo, podemos observar (ver Figura 52) que en el sistema sin control (imagen de la izquierda), el sistema tiende a estabilizarse aproximadamente a los 13 segundos en respuesta a una entrada escalón (línea roja), sin embargo, el sistema no es muy preciso por ende, la respuesta tiende a oscilar.

Mientras que en el sistema con control (imagen de la derecha), la respuesta tiende a estabilizarse en 120 lumens, además, la precisión es la adecuada ya que el sistema tiende a reducir el error (setpoint – valor actual) a un valor aceptable para llegar a un estado estable.

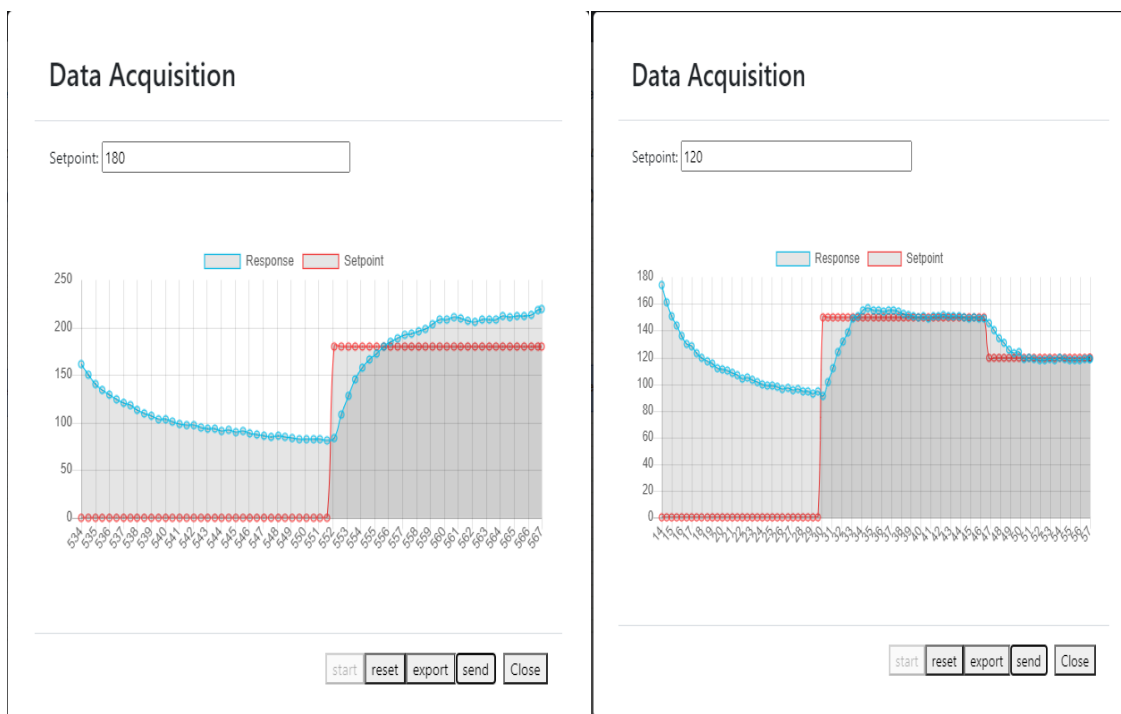


Figura 52. Evaluación sistema iluminación sin control vs sistemas iluminación con control.

La Tabla 1 muestra la evaluación del sistema de iluminación con control según la estabilidad, desempeño y precisión, en contraste al sistema de iluminación sin control.



*Tabla 1 Sistema de iluminación.*

	<i>Sistema sin control</i>	<i>Sistema con control</i>
<i>Tiempo en llegar a un estado estable (desempeño)</i>	<i>15.2 secs</i>	<i>10.8 secs</i>
<i>Rango de variación en lumens (precisión)</i>	<i>0-4 lm</i>	<i>0-1 lm</i>
<i>Estabilidad</i>	<i>El sistema alcanza un punto de estabilidad variando en los rangos de precisión.</i>	<i>El sistema alcanza un punto de estabilidad variando en los rangos de precisión.</i>

La Tabla 2 muestra el promedio en segundos del sistema para llegar a un punto en estabilidad, estos valores se obtuvieron tomando diferentes muestras en el sistema de iluminación.

*Tabla 2 Tiempo de estabilización sistema de iluminación.*

<i>Muestra</i>	<i>Tiempo en segundos</i>
1	12
2	10
3	11
4	12
5	9
Promedio	10,8

Además se evaluó la estabilidad con el mapa de polos y ceros del sistema, como muestra la Figura 53 el sistema presenta un polo negativo en la parte real, por ende, el sistema es estable.

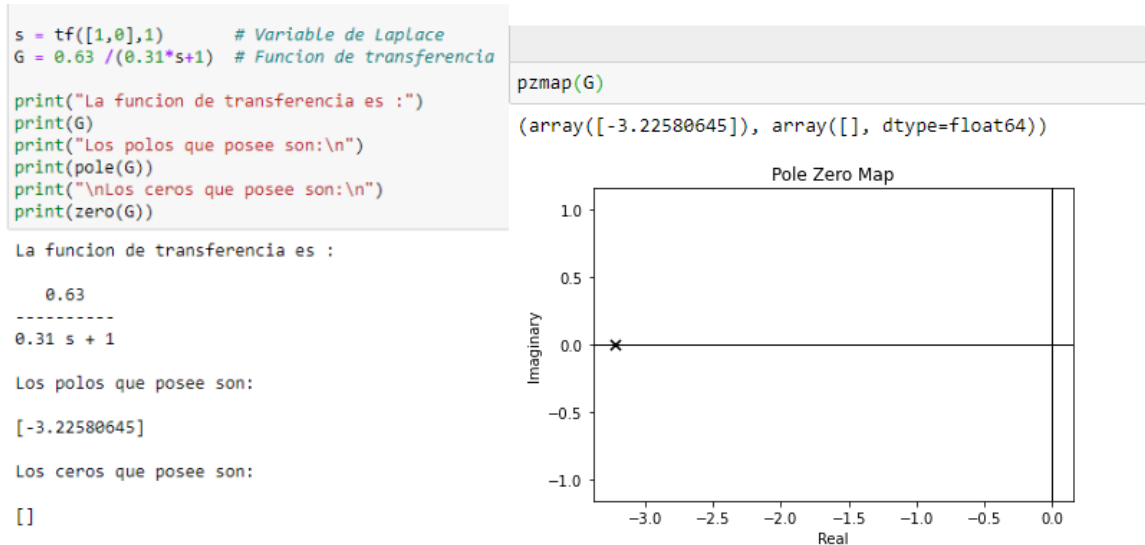


Figura 53. Polos y Ceros en el sistema de Iluminación.

Para el caso del sistema con dinámica discreta (tienda de ropa), en la Figura 54 podemos observar la estimación de la respuesta del sistema con un controlador PID, se simuló datos de las ventas con un comportamiento logarítmico y proporcional añadiendo un controlador PID cuyas ganancias fueron estimadas en el marco de VariaMos.

El sistema se acerca al setpoint deseado y con un rendimiento adecuado, sin embargo, la precisión es el índice a mejorar, al no existir limitaciones físicas como en un sistema continuo, la precisión se ve afectada por restricciones que están en un constante

cambio, por ejemplo, el sistema se puede comportar diferente en los fines de semanas, en una época del año o ya sea por la oferta y demanda que exista en el mercado.

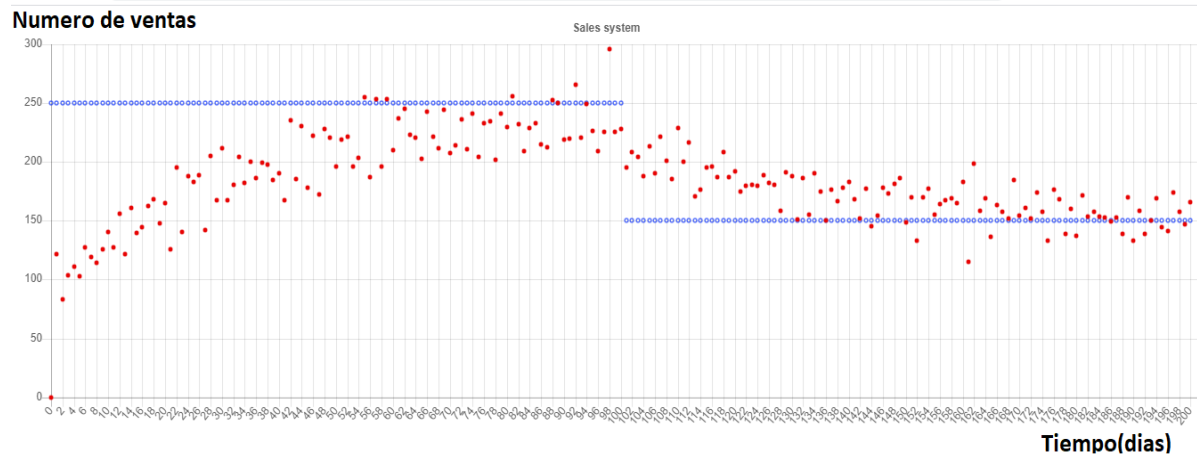


Figura 54. Evaluación sistema de ventas.

## **Capítulo 8**

### **Resultados**

#### **8.1 Módulo de control**

Se implementa el módulo de control en la plataforma de VariaMos el cual permite: el modelado de sistemas de control continuo o sistemas con eventos discretos, estimación de la función de transferencias y las ganancias del controlador utilizando reglas heurísticas desarrolladas por Ziegler Nichols, Cohen Coon y Amigo.

Utilizando la ingeniería dirigida se definió un metamodelo (sintaxis abstracta) y una notación gráfica (sintaxis concreta) para diagramas de bloques, a partir de los cuales se puede producir código fuente Arduino necesario (variables, librerías, métodos) para la aplicación de la teoría de control en diferentes proyectos.

Para facilitar la comunicación entre proyectos Arduino y la plataforma VariaMos se implementa un servicio en el servidor de VariaMos para la comunicación en tiempo real de un proyecto Arduino en el entorno Web de VariaMos.

#### **8.2 Evaluación de un sistema de ventas con controlador**

Para validar el enfoque sistemas discretos se implementa un sistema de ventas con capacidades de control, para la adquisición de datos se simulan ventas con un comportamiento logarítmico o proporcional, posteriormente con la ayuda de la plataforma VariaMos se estima las ganancias del controlador y el modelo matemático del sistema, finalmente, se simula el comportamiento del sistema con un controlador PID.

### **8.3 Conexión Modulo de hardware VariaMos**

En la plataforma de VariaMos diseñada como una herramienta de modelado que incorpora diferentes lenguajes de modelado, se integró la conexión del módulo de control con las capacidades del módulo de diseño de hardware para producir diferentes sistemas con control PID desde cualquier navegador web.

## Capítulo 9

### Discusión y validez de los resultados

Nuestro estudio mostró cómo capacidades como estabilidad y rendimiento de un sistema de control retroalimentado pueden ser acopladas en un sistema de información, el estudio es un paso adelante frente a algunas investigaciones donde se limitaban al desarrollo una aplicación computacional con eventos discretos no necesariamente enfocada a un sistema de información (Leva et al., 2013), ahora la pregunta es conocer si un controlador PID es suficiente en la aplicación de sistemas con eventos discretos más complejos.

Las amenazas a la validez durante la aplicación del enfoque en los casos de estudio de un sistema con eventos discretos se discuten en los siguientes puntos.

**Validez de la conclusión:** el análisis de los resultados depende de los interesados dentro de un sistema de información, por ende, pueden dejar espacio para la subjetividad.

**Validez de construcción:** la principal amenaza para la validez de construcción en el enfoque es el número de perturbaciones en un sistema con eventos discretos, el método se ha definido con la capacidad de adaptación, sin embargo, queda espacio a mejorar en el proceso de precisión del controlador del sistema, a mayores perturbaciones el sistema puede tender a una inestabilidad.

**Validez externa:** las amenazas a la validez externa están centradas sobre posibles factores situacionales adicionales dentro del caso de estudio de un sistema discreto. La aplicación a un solo caso de estudio específico no es suficiente para generalizar los resultados.

## **Capítulo 10**

### **Conclusiones y trabajo a futuro**

Los controles automáticos tienen una intervención cada vez más importante en la vida cotidiana, éstos van desde los simples controles que hacen funcionar un calentador automático hasta los complejos sistemas de control que se requieren para los vehículos autónomos o los sistemas de pilotaje de aviones. Adicional a esto, el control automático se ha vuelto importante para los procesos industriales modernos, por ejemplo, el control de la humedad, la presión, la temperatura, entre otras, en operaciones industriales. Desde el punto de vista de la ingeniería y la ciencia, los sistemas de control automático han brindado los medios para lograr el funcionamiento óptimo de sistemas dinámicos, ya que permiten mejorar la calidad de los procesos, reducir la complejidad asociada a tareas manuales y disminuir los costos de producción. Los sistemas de control se usan con frecuencia en grandes sistemas que tienen componentes de distintos tipos como mecánicos, hidráulicos y eléctricos.

Este trabajo se centró en el control retroalimentado para sistemas con eventos continuos y sistemas con eventos discretos, tanto físicos como de software. Se propone y aplica el enfoque en un caso de estudio de un sistema físico y en sistema de información, la originalidad radica en:

- Proporcionar un proceso de control para sistemas con eventos continuos y discretos.

- Estar diseñado para ser utilizado en diferentes aplicaciones, siempre que incorporen algunos conocimientos mínimos esperados, no necesariamente dirigidos a expertos en control.
- Implementación del enfoque propuesto en la plataforma VariaMos.
- Evaluación del proceso de retroalimentación en un sistema continuo y un sistema con eventos discretos.
- Implementación de un sistema de control basado en la ingeniería dirigida por modelos.

El enfoque permite evaluar un sistema de control retroalimentado en sistemas con eventos discretos. Sin embargo, los sistemas con eventos discretos presentan una gran variedad de circunstancias que deben ser validadas. Además, pensamos en estudios sobre cómo lidiar con el diseño del controlador cuando es necesario controlar un sistema con eventos discretos tipo MIMO (múltiples entradas y salidas) y posibles implementaciones de reglas de ajuste para el controlador en este tipo de sistemas. Este trabajo puede ser extendido para sistemas de información, cuya aplicación no es frecuente, pero sí necesaria.



### Lista de referencias

- Agbesi, R. (2006). *Instrumentation & Control Process Control Fundamentals*.  
[https://www.academia.edu/6067247/Instrumentation\\_and\\_Control\\_Process\\_Control\\_Fundamentals](https://www.academia.edu/6067247/Instrumentation_and_Control_Process_Control_Fundamentals)
- Antonova, M. I., & Batchkova, M. I. (2013). *MODEL DRIVEN DEVELOPMENT OF PROCESS CONTROL SYSTEMS USING UML PROFILES MARTE AND SYSML*. 3, 4.
- Arcelli, D., & Cortellessa, V. (2016). *Challenges in applying control theory to software performance engineering for adaptive systems*. 35–40. Scopus.  
<https://doi.org/10.1145/2859889.2859894>
- Astrom, K. A., & Wittenmark, B. (2011). *Computer-Controlled Systems: Theory and Design, Third Edition* (Edición: Third). Dover Publications.
- Auslander, D. M., Ridgely, J. R., & Jones, J. C. (2017). Real-time software for implementation of feedback control. In *The Control Handbook: Control System Fundamentals, Second Edition* (pp. 16–1). Scopus.  
<https://doi.org/10.1201/b10383>
- Azamfar, M., Markazi, A. H. D., Azamfar, M., & Markazi, A. H. D. (2016). Simple Formulae for Control of Industrial Time Delay Systems. *Latin American Journal of Solids and Structures*, 13(14), 2763–2786. <https://doi.org/10.1590/1679-78253032>
- Badal, F., Das, P., Sarker, S., & Das, S. (2019). *A Survey on Control Issues in Renewable Energy Integration and Microgrid*. <https://doi.org/10.1186/s41601-019-0122-8>
- Berenguer, B. E. P. (2007). *INTRODUCCION A LOS SISTEMAS DE CONTROL*. 69.
- Burghes, D., & Graham, A. (2010). Transfer functions and block diagrams. *Control and Optimal Control Theories with Applications*, 27–47.
- Carrillo, M. (2008). *Ingenieria des sistemas de control continuo ISIDRO LAZARO*.  
[https://www.academia.edu/24667563/Ingenieria\\_des\\_sistemas\\_de\\_control\\_continuo\\_ISIDRO\\_LAZARO](https://www.academia.edu/24667563/Ingenieria_des_sistemas_de_control_continuo_ISIDRO_LAZARO)
- Charfi, A., Li, S., Payret, T., Tessier, P., Mraidha, C., & Gérard, S. (2019). A Model Driven Tool for Requirements and Hardware Engineering. *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 769–773. <https://doi.org/10.1109/MODELS-C.2019.00120>

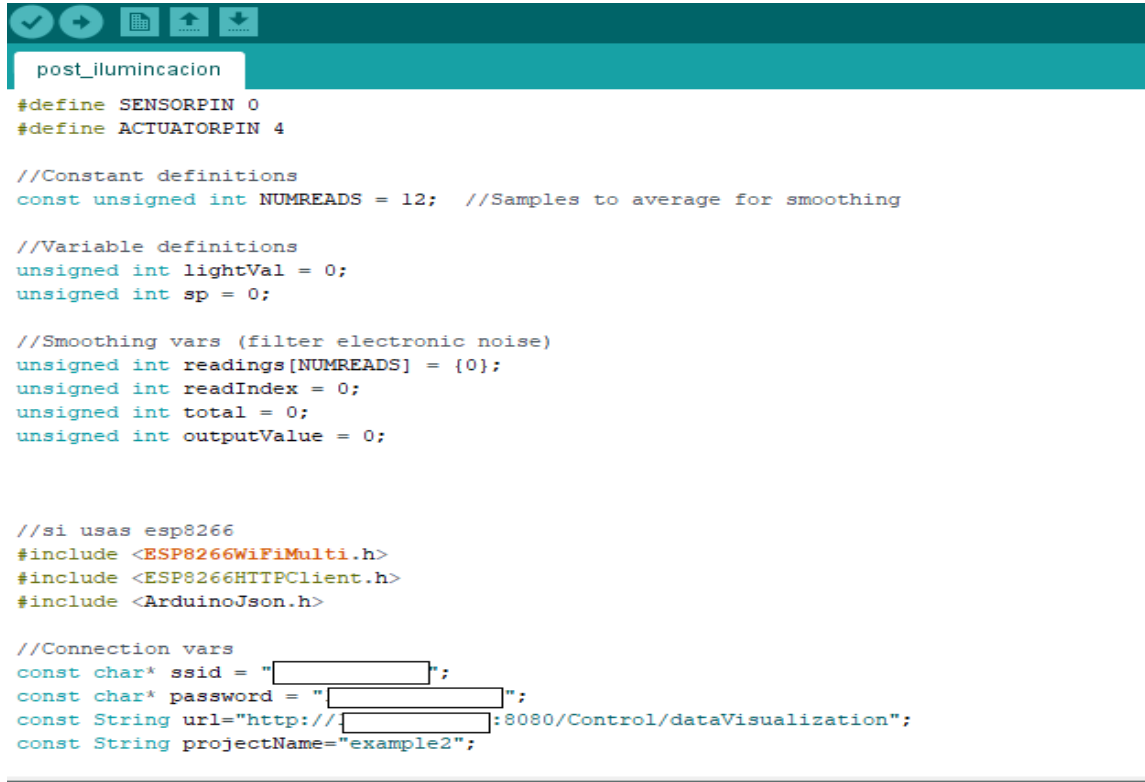
- Christiansen, D., Alexander, C. K., & Jurgen, R. (2004). *Standard Handbook of Electronic Engineering, Fifth Edition with CD-ROM* (Edición: 5). McGraw-Hill Professional.
- Chunjiang Qian, & Wei Lin. (2002). Output feedback control of a class of nonlinear systems: A nonseparation principle paradigm. *IEEE Transactions on Automatic Control*, 47(10), 1710–1715. <https://doi.org/10.1109/TAC.2002.803542>
- Doyle, J. C., Francis, B. A., & Tannenbaum, A. R. (2013). *Feedback Control Theory*. Courier Corporation.
- Frank, S. (2018). *Control theory tutorial: Basic concepts illustrated by software examples*. <https://doi.org/10.1007/978-3-319-91707-8>
- Hagglund, T. (1995). *PID Controllers: Theory, Design, and Tuning* (Edición: 2 Sub). ISA: The Instrumentation, Systems, and Automation Society.
- Hernández, H. G. G. (2015). *Sistemas de control en tiempo continuo y discreto* (Edición: Primera edición). Editorial Digital del Tecnológico de Monterrey.
- Janert, P. K. (2013). *Feedback Control for Computer Systems*. O'Reilly Media. <http://shop.oreilly.com/product/0636920028970.do>
- Jia, Y.-F., Zhou, Z. Q., & Wu, R. (2019). A feedback control approach for preventing system resource exhaustion caused by software aging. *Journal of Internet Technology*, 20(5), 1513–1522. Scopus. <https://doi.org/10.3966/160792642019092005017>
- Kandulna Ruben. (2015). DESIGN OF PID CONTROLLER FOR FOPDT AND IPDT SYSTEM. *NIT Rourkela*.
- Karlsson, N., & Zhang, J. (2013). Applications of feedback control in online advertising. *2013 American Control Conference*, 6008–6013. <https://doi.org/10.1109/ACC.2013.6580779>
- Kawamoto, Y., Yamada, N., Nishiyama, H., Kato, N., Shimizu, Y., & Zheng, Y. (2017). A Feedback Control-Based Crowd Dynamics Management in IoT System. *IEEE Internet of Things Journal*, 4(5), 1466–1476. <https://doi.org/10.1109/JIOT.2017.2724642>
- Kühne, Thomas. (2004). Kühne, T.: Model-driven development: a metamodeling foundation. *IEEE Software* 20(5), 36-41. Software, IEEE. 20. 36 - 41. 10.1109/MS.2003.1231149.

- Kushwah, M., & Patra, A. (2014, June 1). *PID Controller Tuning using Ziegler-Nichols Method for Speed Control of DC Motor*.
- Leva, A., Maggio, M., Papadopoulos, A. V., & Terraneo, F. (2013). *Control-Based Operating System Design*. IET Digital Library.  
<https://doi.org/10.1049/PBCE089E>
- Llobregat, J. J. S., Lacruz, A. C., Calvo, V. C., & Salvador, A. C. (2015). *Control automático. Tiempo continuo y tiempo discreto* (Edición: 1). Reverté.
- Lobontiu, N. (2010). *System Dynamics for Engineering Students: Concepts and Applications*. Academic Press.
- Magalhães, S. C., Fontella, G., Borges, R. F. O., Vega, M. P., Calçada, L. A., & Scheid, C. M. (2020). Construction of software using gain-schedule/auto-adaptive control strategy for automation in drilling fluid production plants. *Journal of Petroleum Exploration and Production Technology*, 10(2), 501–514. Scopus.  
<https://doi.org/10.1007/s13202-019-00746-x>
- Normey-Rico, J. E. (2007). *Control of Dead-time Processes*. Springer Science & Business Media.
- Pascuas, Y., Suarez, J., & Correa, E. (2015). Desarrollo Dirigido por Modelos (MDD) en el Contexto Educativo. *Scientia et Technica*, 20, 172.  
<https://doi.org/10.22517/23447214.9321>
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 45–77.
- Platzer, A. (2012). Logical Analysis of Hybrid Systems. *Logical Analysis of Hybrid Systems*, 7386, 43–49.
- Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software*, 20(5), 19–25. <https://doi.org/10.1109/MS.2003.1231146>
- Sharma, K. L. S. (2011). *Overview of Industrial Process Automation*. Elsevier.
- Simmonds-Mendoza, A., Cabrera-Londoño, N., Berdugo-Barandica, N., Roldán-Mckinley, J., & Yime-Rodríguez, E. (2018). Implementación de control PID de nivel en laboratorio usando PLC Siemens S7-300. *Revista UIS Ingenierías*, 17(2), 159–178. <https://doi.org/10.18273/revuin.v17n2-2018015>

- Simrock, S. (2011). *Tutorial on Control theory*. <https://docplayer.net/28596864-Control-theory-s-simrock-desy-hamburg-germany.html>
- Venkatesan, G. (2002). Modelling feedback control adjustment to control output product quality. *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, 6, 5049–5053 vol.6. <https://doi.org/10.1109/ACC.2002.1025466>
- Visioli, A. (Ed.). (2006). Basics of PID Control. In *Practical PID Control* (pp. 1–18). Springer. [https://doi.org/10.1007/1-84628-586-0\\_1](https://doi.org/10.1007/1-84628-586-0_1)
- Zeqiri, N., & Luma, A. (2008). *Control systems, smart sensors, controller, elements in a control loop, Java Card and Security*. 3.

**ANEXOS**

## ANEXO 1: Código del sistema de iluminación sin control



```
post_iluminacion

#define SENSORPIN 0
#define ACTUATORPIN 4

//Constant definitions
const unsigned int NUMREADS = 12; //Samples to average for smoothing

//Variable definitions
unsigned int lightVal = 0;
unsigned int sp = 0;

//Smoothing vars (filter electronic noise)
unsigned int readings[NUMREADS] = {0};
unsigned int readIndex = 0;
unsigned int total = 0;
unsigned int outputValue = 0;

//si usas esp8266
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

//Connection vars
const char* ssid = "[redacted]";
const char* password = "[redacted]";
const String url="http://[redacted]:8080/Control/dataVisualization";
const String projectName="example2";
```

---

```

unsigned int smooth() { //Recursive moving average subroutine
    total = total - readings[readIndex]; // subtract the last reading
    readings[readIndex] = analogRead(SENSORPIN); // read from the sensor:
    total = total + readings[readIndex]; // add the reading to the total:
    readIndex = readIndex + 1; // advance to the next position in the array:
    if (readIndex >= NUMREADS) { // if we're at the end of the array...
        readIndex = 0; // ...wrap around to the beginning:
    }
    return total / NUMREADS; // calculate the average:
}

void WiFiInit() {
    Serial.print("Connecting...");
    while (WiFi.status() != WL_CONNECTED) { //Check for the connection
        delay(500);
        Serial.print(".");
    }
    Serial.print("Successfully connected, IP adress: ");
    Serial.println(WiFi.localIP());
}

void setup() {
    delay(10);
    Serial.begin(115200);
    pinMode(ACTUATORPIN, OUTPUT);
    digitalWrite(ACTUATORPIN, LOW);
    WiFi.begin(ssid, password);
    WiFiInit();//WiFi communications initialization
}

```

---

```

void postService() {
    HTTPClient http;
    http.begin(url); //Endpoint service
    String json;
    StaticJsonDocument<300> doc;
    doc["times"] = millis()/1000;
    doc["data"] = lightVal;// sensor value
    doc["set"] = sp;// adjustment variable
    doc["name"] = projectName;
    serializeJson(doc, json);
    int codigo_respuesta = http.POST(json); // request

    if(codigo_respuesta>0){
        Serial.println("Code HTTP ► " + String(codigo_respuesta)); //Print return code
        if(codigo_respuesta == 200){
            // server response
            String cuerpo_respuesta = http.getString();
            sp=cuerpo_respuesta.toInt();
            Serial.println("Server response ▼ ");
            Serial.println(sp);
        }
        else{
            Serial.print("Error sending POST, code: ");
            Serial.println(codigo_respuesta);
        }
        http.end(); //freed resources
    }
}

void loop() {

    if(WiFi.status()== WL_CONNECTED){ //Check WiFi connection status
        lightVal = smooth();
        outputValue = sp * 255.0 / 1023.0;
        analogWrite(ACTUATORPIN, sp );
        postService();

    }else{
        Serial.println("WIFI connection error");
    }

}

```



## ANEXO 2: Código del sistema de iluminación con control

```

#define SENSORPIN 0
#define ACTUATORPIN 4

//Constant definitions
const unsigned int NUMREADS = 12; //Samples to average for smoothing

double consKp = 0.4, consKi = 0.2, consKd = 0.0; //Controller constants

//Variable definitions
double sp = 0, y = 0, out = 0; //System Variables

//Variable definitions
unsigned int val = 0;
//Smoothing vars (filter electronic noise)
unsigned int readings[NUMREADS] = {0};
unsigned int readIndex = 0;
unsigned int total = 0;
unsigned int outputValor = 0;

//si usas esp32
#include <HTTPClient.h>
#include <WiFi.h>

//si usas esp8266
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.hpp>
#include <ArduinoJson.h>
#include <PID v1.h>

```

---

---

```

#include <PID_v1.h>

const char* ssid = "[redacted]";
const char* password = "[redacted]";
const String url="http://[redacted]:8080/Control/dataVisualization";
const String projectName="example2";

//Library definitions
PID myPID(&y, &out, &sp, consKp, consKi, consKd, DIRECT);

//Subroutines and functions
unsigned int smooth() { //Recursive moving average subroutine
    total = total - readings[readIndex]; // subtract the last reading
    readings[readIndex] = analogRead(SENSORPIN); // read from the sensor:
    total = total + readings[readIndex]; // add the reading to the total:
    readIndex = readIndex + 1; // advance to the next position in the array:
    if (readIndex >= NUMREADS) { // if we're at the end of the array...
        readIndex = 0; // ...wrap around to the beginning:
    }
    return total / NUMREADS; // calculate the average:
}

void WiFiInit() {
    Serial.print("Connecting...");
    while (WiFi.status() != WL_CONNECTED) { //Check for the connection
        delay(500);
        Serial.print(".");
    }
    Serial.print("Successfully connected, IP adress: ");
    Serial.println(WiFi.localIP());
}

```

---

```

    Serial.println(WiFi.localIP());
}

void setup() {
    delay(10);
    Serial.begin(115200);

    pinMode(ACTUATORPIN, OUTPUT);
    digitalWrite(ACTUATORPIN, LOW);
    myPID.SetMode(AUTOMATIC);

    WiFi.begin(ssid, password);
    WiFiInit();//WiFi communications initialization
}
void postService() {
    HTTPClient http;
    http.begin(url); //Endpoint service

    String json;
    StaticJsonDocument<300> doc;
    doc["times"] = millis()/1000;
    doc["data"] = y;
    doc["set"] = sp;
    doc["name"] = "proyect2";
    serializeJson(doc, json);

    int codigo_respuesta = http.POST(json); // request

```

---

```

if(codigo_respuesta>0){
    Serial.println("Code HTTP ► " + String(codigo_respuesta));    //Print return code
    if(codigo_respuesta == 200){
        // server response
        String cuerpo_respuesta = http.getString();
        sp=cuerpo_respuesta.toInt();
        Serial.println("Server response ▼ ");
        Serial.println(sp);
    }

    }else{

        Serial.print("Error sending POST, code: ");
        Serial.println(codigo_respuesta);
    }
    http.end();    //freed resources
}

void loop() {

    if(WiFi.status()== WL_CONNECTED){    //Check WiFi connection status
        y = smooth();
        myPID.Compute();
        analogWrite(ACTUATORPIN, out );
        postService();
    }else{
        Serial.println("WIFI connection error");
    }
}

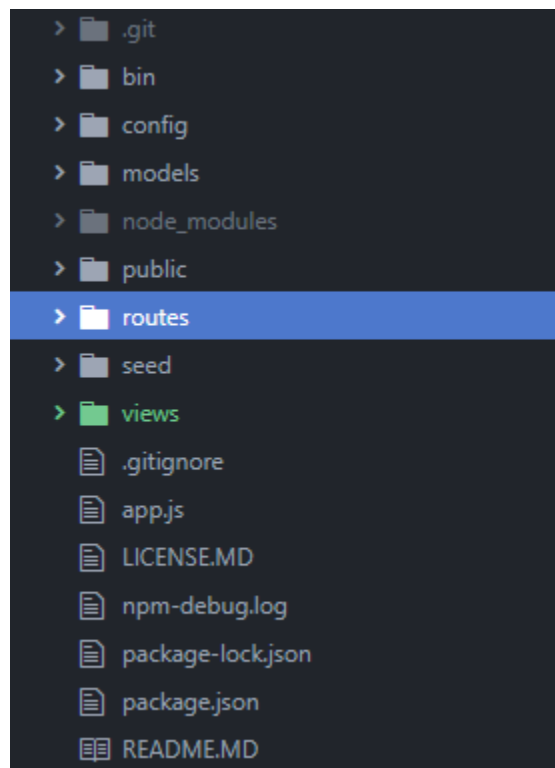
```

---

## ANEXO 3: Código fuente tienda de ropa

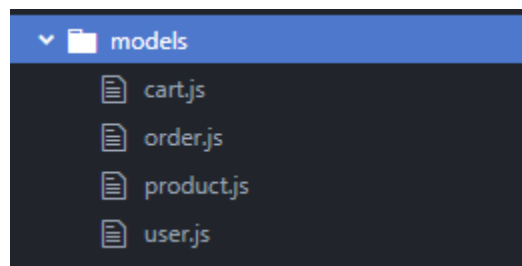
### 3.1 Estructura de carpetas

A continuación, se presenta la distribución de carpetas del proyecto, el código fuente está disponible en <https://github.com/andres4588/Shopping>.



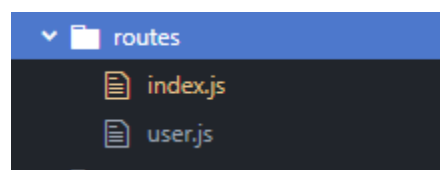
### 3.2 Modelos

En la carpeta **models** se encuentran los datos de la aplicación como información de los productos, órdenes, usuarios, ficha de producto (producto seleccionado, precio y cantidad seleccionada).



### 3.3 Controladores

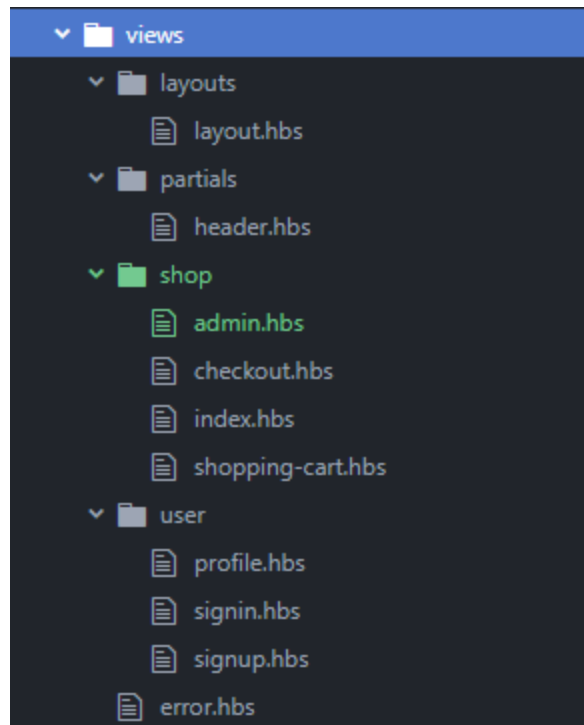
La carpeta **routes** contiene los controladores que permiten comunicar los modelos con las vistas en la aplicación, se tiene un controlador para manejar toda la información de las ventas (index.js) y otro para manejar el proceso de autenticación de un usuario (user.js)



### 3.4 Vista

En la carpeta **views** se encuentra la interfaz de la aplicación, la carpeta **layouts** se encuentra el esqueleto de la página, la sección de **partials** contiene el encabezado de la

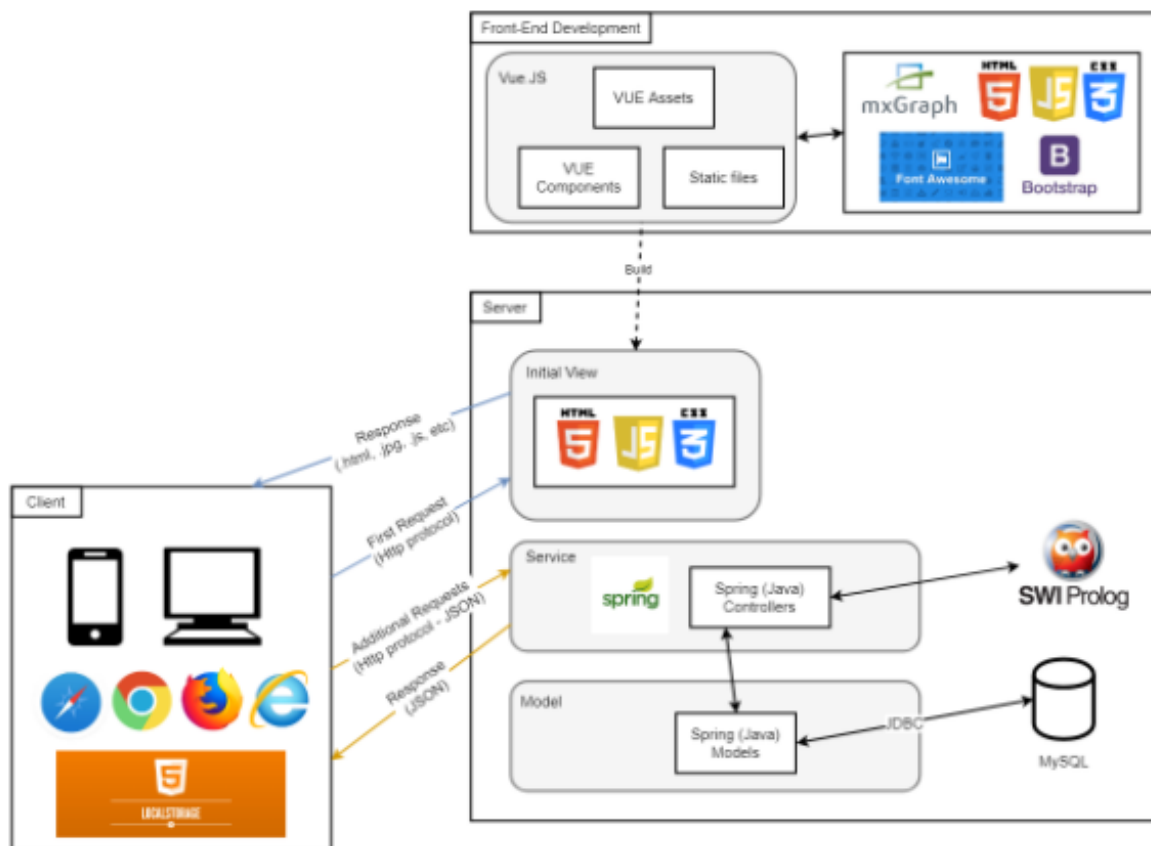
página, la carpeta **shop** muestra las plantillas para el proceso de ventas y la carpeta **user** las plantillas utilizadas para el proceso de autenticación,



## ANEXO 4: VariaMos Web

### 4.1 Arquitectura

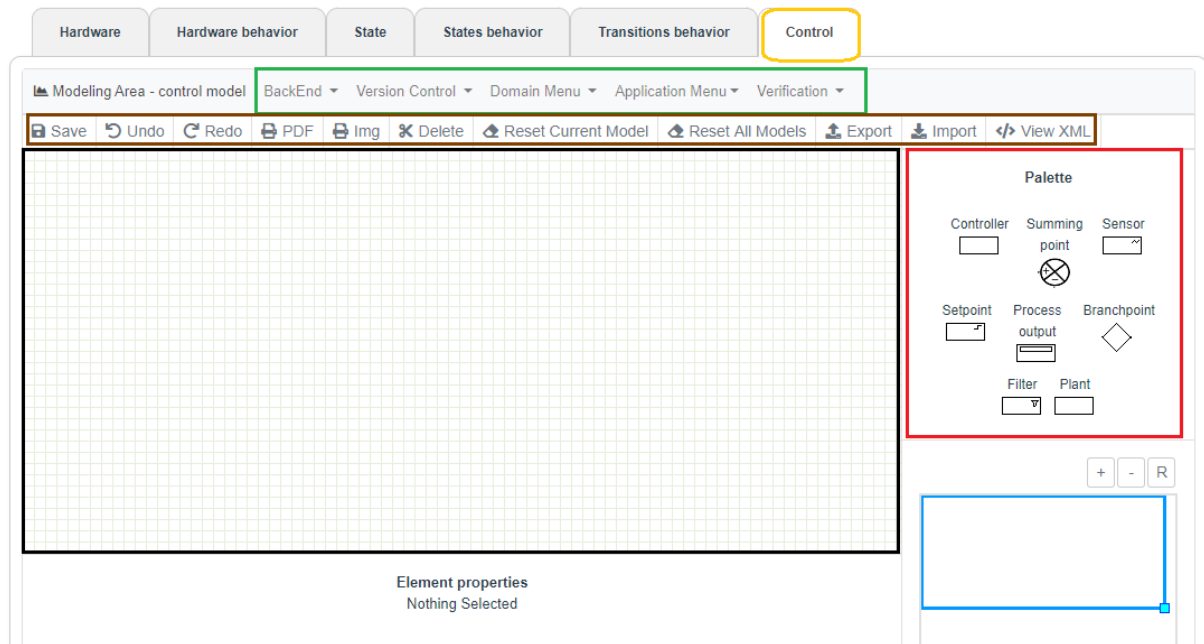
La arquitectura web de VariaMos se divide en tres vistas principales. Vista de desarrollo de front-end, vista de servidor y vista de cliente.



#### 4.1.1 Vista del Cliente Modulo de control

Variamios fue diseñada como una herramienta de modelado que incorpora diferentes lenguajes de modelado, en este trabajo se desarrolló un nuevo módulo para el diseño de sistemas de control.





En esta se pueden observar 5 grandes áreas:

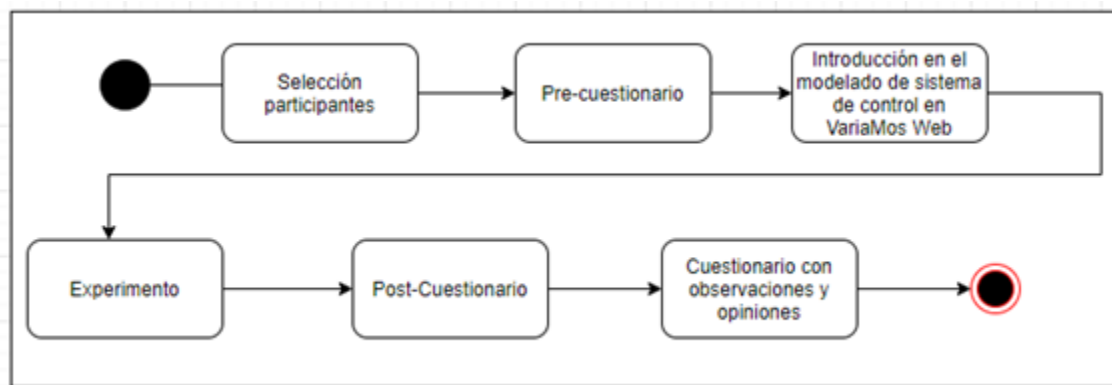
1. En amarillo tenemos la pestaña correspondiente al modelo de control.
2. En negro tenemos el área de trabajo, aquí es donde podremos arrastrar los elementos desde la paleta.
3. En verde se encuentran diferentes opciones para el modelo de control ya sea visualización de datos, generación de código arduino o configuración del backend, en las siguientes secciones se mostrará en detalle el uso de estas opciones.
4. En marron.es la barra de herramientas, podemos encontrar algunas funcionalidades como importar o exportar el modelo, con la barra de herramientas estaremos guardando el modelo o borrando todo en el área de modelado.

5. En rojo encontramos la paleta, en el cual se definen cuáles son los elementos permitidos para crear su modelo

## ANEXO 5: Prueba de usabilidad

En esta sección se presenta una prueba de usabilidad del enfoque de control en sistemas discretos y en sistemas continuos en VariaMos. El objetivo es recolectar información sobre la facilidad de uso del enfoque para el desarrollo de sistemas de control.

La prueba de usabilidad está dividida en 6 partes descritas a continuación.



1. **Selección de participantes:** Se realizará una selección de mínimo ocho estudiantes (de acuerdo con las recomendaciones del estándar ISO/IEC 25062 para reportar test de usabilidad) de la universidad EAFIT del programa Ingeniería de Sistemas o afines interesados en probar la usabilidad del control de sistema en VariaMos Web. La prueba de usabilidad fue diseñada para realizarse en una sesión de aproximadamente 60 minutos. Los participantes se clasifican como “Estudiantes de ingeniería de sistemas o carreras afines.

2. **Pre-cuestionario (5 minutos):** Se solicita a los participantes que completen un cuestionario relacionado con su experiencia previa en el desarrollo de sistemas de control. El cuestionario se elaboró utilizando una escala Likert: 1) muy de acuerdo, 2) algo de acuerdo, 3) ni de acuerdo ni en desacuerdo, 4) algo en desacuerdo, y 5) muy en desacuerdo.
3. **Introducción a Variamos web enfocado en el modelado de sistemas de control (5 minutos):** Se realiza una presentación introductoria de los conceptos claves de un sistema de control, además, se presentan los componentes disponibles en la plataforma VariaMos Web, esta introducción es fundamental para que los participantes adquieran el conocimiento básico de la plataforma VariaMos Web y cómo usarla para el modelado de sistemas de control.
4. **Experimento (35 minutos):** Se le pedirá al estudiante que realice el experimento completar una secuencia de actividades; quien dirige el experimento atenderá las dudas que le surjan al estudiante durante el experimento.
5. **Post-Cuestionario (5 minutos):** Los participantes responderán un cuestionario, el cual incluirá preguntas sobre: i) actores externos, ii) la satisfacción general, iii) el rendimiento de Variamos Web enfocado en el control de sistemas y un apartado para comentarios adicionales.

6. **Cuestionario con opiniones y observaciones (5 minutos):** Se formularán al participante cinco preguntas abiertas sobre la usabilidad de la herramienta y se recibirán comentarios y observaciones relacionados con la experiencia de uso de Variamos Web.

### 5.1 Métricas

Para evaluar la usabilidad del enfoque se definieron tres métricas: efectividad, operabilidad y satisfacción.

Para medir la efectividad se tienen en cuenta los siguientes ítems:

- **Tareas completadas:** número de tareas completadas correctamente.
- **Tareas no completadas:** número de tareas no completadas correctamente.
- **Ayuda:** número de veces que el participante solicita ayuda verbal para completar una tarea.

Para medir la operabilidad se registraron los siguientes ítems:

- **Claridad del mensaje:** (proporción de mensajes que son explicados por sí mismo) evaluado por la pregunta “Los mensajes (descripciones, títulos) en el modelo de control fueron claros y precisos” realizada en el post-cuestionario.
- **Claridad del elemento de la interfaz:** (definido como la proporción de elementos de la interfaz son explicados por sí mismo) evaluado por la pregunta “Los elementos de la interfaz en el modelo de control fueron claros.” realizada en el post-cuestionario.

- **Entrada y salida comprensibles:** evaluado por la pregunta “Los datos de entrada requeridos y los datos de salida proporcionados fueron claros” realizada en el post-cuestionario.

Para la satisfacción, se utilizó los resultados del post-cuestionario con el fin de medir la satisfacción subjetiva del estudiante en relación con la herramienta, además se incorpora los resultados del cuestionario con opiniones y observaciones para una mejor percepción del estudiante.

## 5.2 Resultados

### *5.2.1 Efectividad*

Como resultado de la métrica de efectividad los participantes de la prueba de usabilidad completaron correctamente las 7 tareas asignadas, algunos estudiantes necesitaron de ayuda verbal para terminar unas tareas como se observa en la siguiente figura

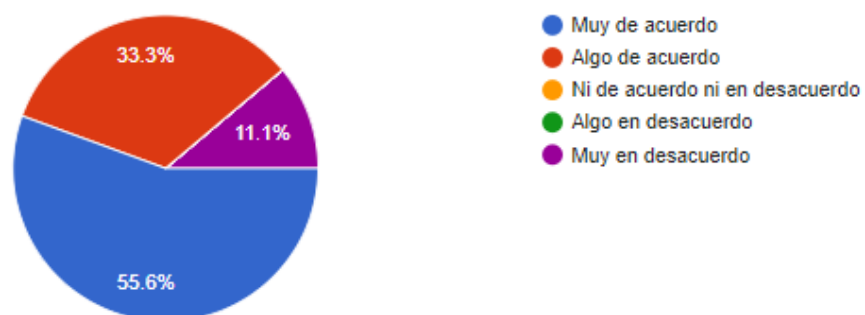


## 5.2.2 Operabilidad

### 5.2.2.1 Claridad del Mensaje

Los mensajes (descripciones , títulos) en el modelo de control fueron claros y precisos.

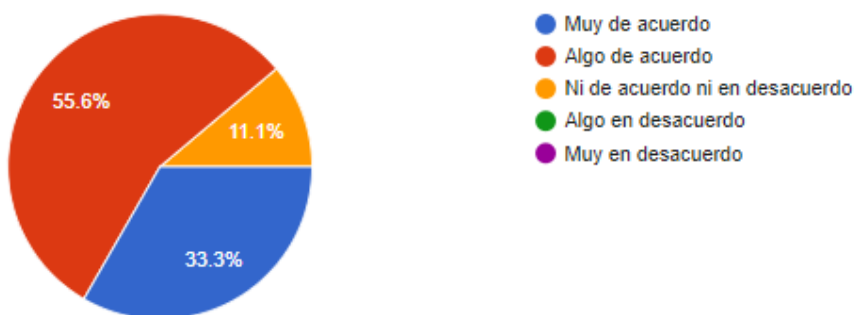
9 respuestas



### 5.2.2.2 Claridad del elemento de la interfaz

Los elementos de la interfaz en el modelo de control fueron claros.

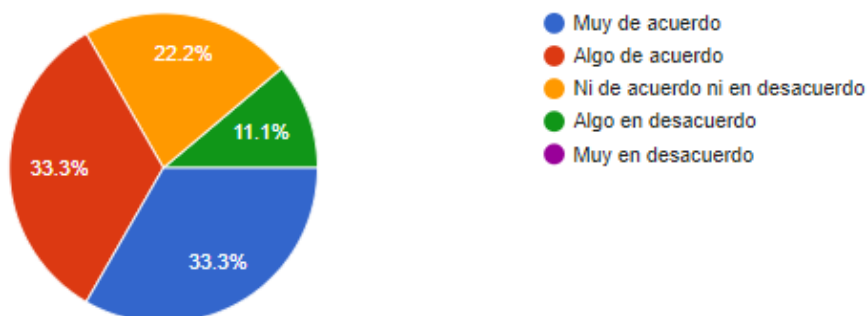
9 respuestas



### 5.2.2.3 Entrada y salida comprensible

Los datos de entrada requeridos y los datos de salida proporcionados fueron claros?

9 respuestas



### 5.2.3 Satisfacción

Finalmente, para la satisfacción en general a los estudiantes les gusto el modelado de sistemas de control en VariaMos Web, describiendo que es una herramienta útil para



personas no expertas en control y que necesitan diseñar controladores sin tener conocimientos avanzados en la solución de ecuaciones diferenciales.

Algunas recomendaciones o sugerencias para mejorar la herramienta son:

- Utilizar colores más contrastados.
- Podría incluirse el tutorial dentro de la plataforma en un menú de ayuda
- Modificar algunos rectángulos para que sea más fácil recordarlos pueden ser etiquetas
- Añadir una opción de información donde se detalle cada una de las opciones en VariaMos.